ATARI CLASSICS

The Magazine for the Dedicated 8-Bit User

Epson Inkjet/Laser
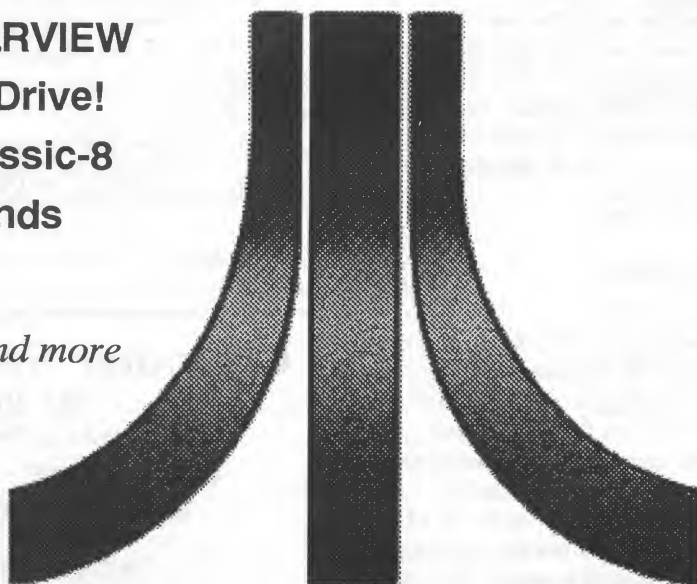Printers for the 8-Bit:
Two Articles!

In this Issue:

The Last Word on 8-Bit DTP
MIDI for The Classic-8
Genlock for the 8-Bit
More About COLRVIEW
Fix your Floppy Drive!
SCSI On The Classic-8
Subscriber Refunds
Tips 'n' Tricks

...and more

Our Final Issue (??)

Table of Contents
Vol. 3 No. 2 April 1994

FEATURES

COLUMNS

DEPARTMENTS

From The Editor's Desk...

Closing Up Shop

by Ben Poehland, AC Managing Editor

A Promise Is A Promise

In my February '94 editorial, I stated that *AC*, unlike other Atari publications in the past, intended to compensate its readers for the unused portions of their subscriptions as we bring this unusual experiment to a close. At the time I made that commitment I had a gut feeling it would be one of those things that's easier said than done. I was right.

Working out a policy to treat our loyal readers as fairly as possible within the limits imposed by cost was a thorny dilemma.. A lot of time was spent consulting with some of the *AC* Staff, the postal authorities, and especially our publisher to figure out how to handle the situation. For better or worse, I finally arrived at a set of policies to cover all contingencies and still keep our objective in focus.

That objective is to wind down this operation with as little financial burden to our readers as possible, and to be as fair about it as we can within the constraints of cost. There are some broad philosophical concepts I employed to guide the formulation of policies for issuing refunds and closing up shop. I'll share these with you, so you'll get a feel for the complexity of the problem I faced and know what reasoning went into the policies Unicorn and I finally agreed upon.

General Principles

When dealing with any organization whose resources are limited and whose services are needed by many, I followed the principle that those whose demands are least should be satisfied first, for in this manner the greatest number of people will be satisfied. Application of this principle to *AC* resulted in a prioritization of subscriptions into two broad classifications: those inside the USA and those outside the USA. American subscribers constitute 85% of *AC*'s reader base, an overwhelming majority. Overhead costs for issuing refunds to those readers are also the lowest. Consequently, American readers (subject to a cost cutoff explained below), are awarded highest priority in the issuing of refunds.

Of the remaining 15% of our paid subscriber base, roughly 5% are Canadian while the other 10% originate from a variety of countries scattered all around the globe. By taking advantage of the Canadian banking system's liberal policies in the treatment of US funds, the issuance of refunds to Canadian subscribers ends up being only slightly more burdensome than refunds made to American subscribers. Consequently Canadian readers are awarded the middle level of priority for issuance of refunds. These first two categories of priority thus provide for 90% of our readers.

The final 10% of our readers who are outside North America are regrettably assigned lowest priority. They constitute a small fraction of our readers, and reimbursement costs to them are the highest. Most of these subscribers are in Britain and Europe, plus a few more in the Pacific (Australia and New Zealand). Whether or not these subscribers receive a refund will depend upon a variety of factors, the most important of which is the method by which they originally paid for their subscription. Credit card subscribers will be assigned a higher priority within this group than those who paid by bank draft or money order.

Very broadly, the prioritization of subscriptions as described above, in concert with the policies already in effect (as announced in my editorial in the February '94 *AC*) is envisioned to produce the following results. A small percentage of our subscribers, about 10%, (labels bearing 0294), will receive an extra magazine they didn't pay for. Another small group, about 10-15% of our subscribers, will lose a small amount of money. The amount will vary from about $5.00 up to about $12.00, depending upon the classification of their subscription. And finally, the great majority of readers, 75-80%, will receive a full refund and break even. Although there are bound to be a few cases of inequity, overall things should balance out to neutral. That's assuming, of course, that the magazine's financial assets are sufficient to cover all categories of refunds.

No Guarantees

At the moment I honestly don't know *AC*'s exact financial status. Nor is it possible to determine what the magazine's financial status will be until we actually cease publication. So there's no guarantee we'll be able to issue refunds to everyone who qualifies for one.

At the time of this writing (early April), *AC* is still a fully functional business enterprise. Although its income has dwindled drastically in 1994, there is still some income (mostly from advertising revenue). And of course we have yet to pay the full production expenses for the publication of this final issue. Since our final income hasn't yet been tallied, and our final production expenses haven't yet been paid, we don't know exactly how large will be the remaining pool of funds from which reimbursements will be made. And the biggest unknown of all is what the sum total of all the refunds will add up to.

In short, the actual processing of refunds is something that will play out as we go. We'll simply commence issuing refunds to everyone who qualifies for one, until everyone is paid or the money runs out-- whichever happens first. The order in which refunds are made will be determined by the general priorities described above. My commitment to issue refunds is

based on estimates of *AC*'s financial condition furnished me by Unicorn Publications. I know that substantial resources exist. What we can't predict is exactly how far those resources will stretch.

Cost Cutoffs

In discussing the cost of making refunds with our Publisher, it quickly became apparent there was a certain level of reimbursement below which it simply was not cost-efficient to issue a payment, due to the overhead cost involved. The cutoff figure we agreed upon was $5.00. What this means is, if the calculated value of your refund amounts to less than $5.00, you won't receive a refund. For non-USA subscribers, this cutoff is higher due to the extra expenses of currency conversion, credit card transaction fees, and money order fees. A cost cutoff will be applied to all classifications of subscription refunds.

At first glance the imposition of a cost cutoff may seem unfair. In practice, however, it will only affect a relatively small number of subscribers. By applying the cutoff we are ensuring a larger pool of funds available to the many who are due substantial amounts of money, at the expense of a few who are due trivial sums at high overhead cost. It is consistent with the general principle of conserving our resources to ensure most of the money we send out ends up in the hands of subscribers instead of bank fees or postal agencies.

Practically speaking, application of the cost cutoff means that if the expiration on your mailing label is 0694 (and you didn't subscribe to the Disk), you won't get a refund. You *might* get a refund if you subscribed to the disk or your label expiration reads 0894. If your label reads 1094 or 1294 (the majority of readers), you'll almost surely receive a refund—subject to the limits of the available pool of reimbursement funds. And of course, labels bearing 0294 or 0494 are subscriptions that have already expired, so those readers won't even be considered for a refund.

Are you confused yet? Welcome to the club! If what I've presented thus far has made for difficult reading, be assured that making all these decisions and trying to describe it hasn't exactly been a picnic. Perhaps things will come into sharper focus when you see the detailed summary that follows.

Refunds In Detail

Prioritized classes of refunds and cost cutoffs are as follows:

A.) **USA subscribers**. Anyone who is due a refund of $5.00 or more will receive a check. If the amount of the refund totals less than $5.00, no refund will be issued. Class A subscriptions will be processed first.

B.) **Canadian subscribers**. Anyone who is due a refund of $8.00 or more will receive a check in U.S. funds. Most Canadian banks routinely handle drafts in US$ and typically charge $3.00 for currency conversion, which cost will be borne by the subscriber. If the amount of the refund totals less than $8.00, no refund will be issued.

C.) **Non- North American subscribers who paid by credit card**. Anyone who is due a refund of $10 or more will have a credit issued to their charge account, provided the account is still current. A $2.00 transaction fee will be charged by the publisher, to be borne by the subscriber. Currency conversion fees charged by the credit card issuer will also be borne by the subscriber. If the charge account has expired, *AC* will attempt to reimburse the reader according to classification D below. If the amount of the refund totals less than $10, no refund will be issued.

D.) **Non-North American subscribers who paid by bank draft or IMO**. Included in this category also are readers whose credit account has expired since they subscribed. Anyone who is due a refund of $12.50 or more will receive a refund via International Postal Money Order, the $7.50 IPMO fee to be borne by the subscriber. If the amount of the refund totals less than $12.50, no refund will be issued. Class D subscriptions will be processed last.

Logistics

Atari Classics will bear the overhead costs of check processing, postage and paper. We presently anticipate issuing around 300 refunds. Credit card transactions and IPMO payments will originate from Unicorn Publications in Michigan, while the actual mail-out of credit slips and and checks will be done from my Pennsylvania location from bulk shipments of same sent to me by our publisher. The internal logistics are necessarily clumsy because the magazine's distribution and accounting functions aren't all under one roof, so they've never been integrated.

Who will determine your refund eligibility and amount? I will. Once the distribution of this final issue is completed, I'll be combing the subscriber list one entry at a time, applying the policies outlined above, and submitting lists of names and amounts to our Publisher. Within each category, refunds will be issued in alphabetical order by last name. No one associated with the magazine Staff, including me, will receive preferential treatment. If the cash flow runs dry just as I get to the name of some Editor or Columnist or whatever, tough krinkles guys. As subscribers we all stand equal.

Having never done anything like this before, I have no idea how long the process will take. I'd like to see all the magazine's accounts settled by August 1, 1994, but that might be wishful thinking. All inquiries from anxious readers will be ignored; I won't have time to answer volumes of frivolous mail and work on refunds as well. I think it's safe to say that if you haven't received a refund by September 1994, you aren't going to get one. No amount of desperate begging/pleading will help: our refund policies are carved in stone.

For those of you who subscribed to *AC* late in its existence and don't have a full complement of issues, I'll continue to make Back Issues of magazine and Disk available as long as my stock permits. Please refer to the masthead on page 3 for policies relating to the purchase of Back Issues. I especially caution readers NOT to request their refunds in the equivalent value of Back Issues. Refund payments are handled by Unicorn Publications, while Back Issue sales are

processed personally by me—and ne'er the twain shall meet.

What Now?

On page 15 of this issue you'll find an enticing editorial by Jim Hood detailing the intention of our California-based colleagues to keep the *Atari Classics* tradition alive. I still can't decide if Jim Hood and Bob Woolley are the bravest fellows I've ever met, or if they simply escaped from the same cell at the local asylum and haven't been rounded up yet. (As regards the latter possibility, I should perhaps not speak so boldly, as there have been enquiries about my *own* sanity of late.)

No matter, as long as there is any kind of effort whatever, I'll certainly support it. I'm favorably impressed with the flavor of what I see on pages 15-18. I'll not wait to receive my own refund as Jim suggests, but will forthwith send my new subscription to California as soon as I receive my copy of the April '94 *AC*. And I encourage all of you to do the same if you wish to see our community, our market, and the usefulness of our machines preserved. I do honestly believe this will be our last stand, our final opportunity for existence as a recognizable entity. If the California effort fails, you might as well pack away your 8-bit and start reading up on MS-DOS.

Mainstream Perspectives

The direction of the mainstream computer market is heavily slanted toward the "messy-DOS" machines anyway, and old PC/XT/AT machines are to be had almost for the asking (I've acquired a closetfull of them, 9 at last count). For anyone who remembers the "computer wars" of 1984, when the giants of the home computer industry battled for supremacy, it's now clear the Intel/Microsoft architecture has won. Even mighty Apple—whose founders trace their beginnings to the old Atari under Nolan Bushnell—is straining to maintain its 10% niche of the mainstream market. And Atari Corp., despite the introduction of XL's, XE's, ST's, STe's, Megas, Falcons, Lynxes, and Jaguars, doesn't even make a dent in the mainstream computer world. Or the videogame world any more either, for that matter.

After 10 years of using my trusty 800XL, I find myself sucked irresistibly toward the mainstream computing world. Partly this is due to the demands of my job: nobody with an active career in science can afford to ignore trends in computing technology. This attraction has always existed, yet until lately I've successfully resisted it. It isn't so much that the lure of the PC/Mac has grown stronger; rather, it's the growing inability of my trusty 800XL to serve my changing needs, which are all software-driven.

I'm much impressed with Dave Richardson's rendering of laser-generated Daisy-Dot 3 output in this issue. It's very reassuring to know there's at least one 8-bit DTP program that can benefit from laser printing technology without modification. However, I fear Dave's success at coupling DD-3 to a laser printer is more the exception than the rule.

As I read over Jeff McWilliams' "Moonlight Workshop" column in this issue, it struck me that Jeff was again telepathically tuning-in to my private thoughts. The tone of frustration implicit in his experience with the Epson Stylus 800 is a microcosm of a larger problem that, if not soon remedied, spells final doom for us devotees of the classic Atari. The problem is lack of software, especially productivity-oriented material capable of interfacing with today's sophisticated hardware technology.

Software: The Final Frontier

The actual "machine" we call a computer is purely a hardware contrivance. But the sad fact is, in the computer world hardware no longer counts for much. The difference between a computer that gets used and one that doesn't is the availability of software. Folks, take a look around. Shareware and PD offerings notwithstanding, there just isn't enough input on the software end of our market to keep our hardware viable. Advances in hardware technology continue to occur at a dizzying pace. As Dave Richardson and Jeff McWilliams have demonstrated, our Classic-8's are capable of utilizing at least some of this advanced technology, if only someone will write the software.

Our existing software base is just plain decrepit. Much of the commercial software available for our machines is woefully outdated and needs to be debugged or rewritten to take advantage of newer hardware and more powerful DOS's. (The recent effort of Fine Tooned Engineering to spruce up the code in some of the old OSS/ICD products is a step in the right direction.) Perhaps the old stuff is sufficient for some users, but for those of us whose needs are still evolving, hanging onto the beloved Classic 8-bit is becoming a process of slow strangulation.

A case in point is software for generating camera-ready electronic diagrams (such as appears in Paul Alhart's "A-T-A-R-I" column elsewhere in this issue). With today's 24-pin, inkjet, and laser printers, an XL/XE machine with expanded memory is entirely capable of generating satisfactory schematics. Instead, what do we have? Graphic Master, Circuit Database, and Schematic Designer: all of them outdated, amateurish, inadequate, and unuseable for any serious application. I don't mean to scandalize anyone, but even Bob Woolley (as fanatical a devotee of the Classic-8 as I am), used Schema on an IBM to generate the schematics for his December TTL video upgrade article. If Bob could have generated schematics of equivalent quality on his beloved 1200XL, you can bet he would have done it. But there just isn't a Schema for the XL.

The cruelest hoax I ever experienced was Graphic Master's claim to be able to produce "chemistry symbols". The first time I used the program it became painfully obvious no one at DataSoft had ever bothered to ask a real chemist what was required of such software (the chemistry symbol library was a joke). If they had, they'd today be competing with programs like ChemDraw for the Macintosh, which sells for $900 a copy. The graphic below, which I created in ten minutes on a $3,000 Mac IIci with a $2,000 Hewlett-Packard laser printer, could have been rendered reasonably

well on my 1-meg 800XL with a 24-pin printer
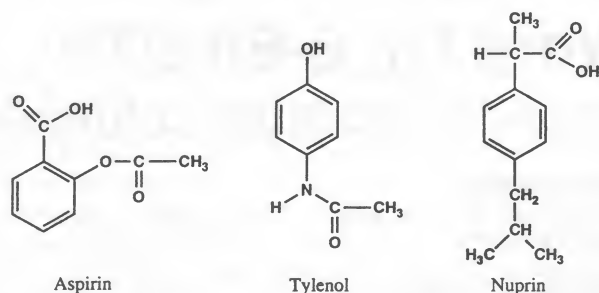
Figure 1. Chemical Structures of Common Pain Relievers

if the software existed. Sadly, my employer forced me to take my 800XL home after five good years of service because I couldn't draw chemical structures with it. It was software, not hardware, that turned my 800 XL out to pasture.

The Mars-8 product now in development at Fine Tooned Engineering is rumored to provide a staggering 4-megs of expanded memory. But, what software will utilize it? Atari gave us a nice 80-column hardware upgrade, the XEP80, but you don't even need all the fingers of one hand to count the software applications written for it (and *none* of those are graphics programs). The most technically advanced computer hardware is so much worthless junk without correspondingly powerful software to complement it.

800XL Or IBM?

I walk into one of the many warehouse-type computer stores springing up all over my area and see endless aisles of software for the PC, including some very attractive graphics packages. A day will come when I'll find PC programs that'll permit me to draw electronic diagrams and chemical structures. There being already tons of programs for the IBM to do the routine stuff I'm already doing with my 800XL (word-processing, telecommunications, and databases), a few more quick PC software purchases will complete the obsolescence of my faithful 8-bit.

That day is coming; I'm not sure when. For the present I'll continue to pound this same keyboard I've beaten to death for the past 10 years (the textured surface on the keys is now worn completely smooth) on an 800XL which is as familiar to me- inside and out- as my favorite pair of old shoes. And I'll continue to support *Atari Classics* for as long as it, and I, occupy the same sphere of interest.

One Last Salute

Before signing off this final missive in my capacity as Managing Editor of the AC Experiment, I'm compelled to pay homage to the Columnists, Editors, Reviewers, Authors, and Subscribers who all contributed to the enviable tradition established by this publication during its all-too-brief existence. Amateurs we may be, but I'm satisfied we gave a good account of ourselves. We sail forward into an uncertain future secure in the knowledge we did our best.

Tips 'n' Tricks

Gigabytes For The 8-Bit—For the truly power-hungry Atari 8-bit user who owns the CSS Black Box: where the BB documentation refers to dividing a hard drive into 99 separate AtariDOS 707-sector partitions, it should also mention that each of those 99 partitions can be up to 16 megabytes each if you're using SpartaDOS or MyDOS. That gives our lowly Atari 8-bit machines the ability to utilize up to 1.6 gigabytes of hard drive space. IBM, eat your heart out! (Thanks to columnist Paul Alhart for this tip.)

Poor Man's Spike Protector—If you've ever experienced the agony of a power-surge "glitch" on your AC line causing all manner of havoc with your computer, you might want to try the following fix which several pundits claim is effective and costs you nothing. Just tie a "granny" knot (a single loop, once over-under-around-and-through) in your line cord. One knot at the plug, and for good measure another where the line cord enters the power supply. It probably wouldn't hurt to have a knot or two in the DC line between the power supply and computer, either. Pull these knots as tight as you can. Do it for all your peripherals, including disk drives, monitors, printers, and modems. Of course this isn't as effective as a real line-suppression module, and nothing will protect you against a direct lightning strike. But it definitely will buy you some measure of protection, you got nothing to lose. This is one of the rare occasions when you can play the laws of physics to your own benefit and actually get something for nothing.

D-Y-I 64K BB Print Buffer—If you own a CSS Black Box sans printer buffer, Bob Puff has provided the following information to allow you to add the 64K printer buffer option as a do-it-yourself project. This information was provided by CSS with the proviso that CSS not be held liable for any Black Box that gets damaged if you muff it. (*AC* won't be responsible, either.) The mod is reasonably inexpensive, but does require some dexterity with a small soldering iron and experience working with microchips. If you have a Black Box ROM Version 1.xx release, this modification *will not work*. Contact CSS (see their ad in this issue) for info on obtaining upgraded BB ROMs.

You'll need two 32Kx8 Static RAM chips (SRAMs): Sony CXK58256PM-12 or NEC D43256AC-12 or equivalent. Jameco Electronics sells both chips, although the NEC chip is cheaper-- $5.75 in the July 1993 catalog. Contact Jameco toll-free at 1-800-831-4242 or write to: 1355 Shoreway Road, Belmont CA 94002 USA. Beware of a salesman who might try to steer you toward a faster SRAM chip used as "cache RAM" in MS-DOS machines, such a 61C256. These are the wrong size, among other things. You must have a 28-pin DIP that matches the 8Kx8 SRAM already installed in a stock BB (a Hitachi HM6264LP-15). You'll also need a few lengths of insulated wire and the usual assortment of small hand tools to complete the job.

Stack the two SRAMS and solder all the pins together except pins 20 and 22. Raise pins 20 and 22 of the upper SRAM and tie them together. Remove the 8K SRAM from the Black Box and install this stack, insuring that the index notch in the end of the chips is pointing in the same direction as the original SRAM.

Now locate the 74LS139 chip in the front right-hand corner of the Black Box. Solder a wire between pin 5 of this chip and pins 20 and 22 of the SRAM stack. Lift pin 3 of this same LS139 chip and solder a wire between it and pin 9 of the 6821 chip on the Black Box (a 40-pin chip to the left of the SRAM stack). That's all there is to it! Reconfigure your Black Box by going to the Port Statuses screen and changing the Spooler Stat line to read Internal RAM: your 64K printer buffer is now active. (Thanks to *AC*'s Charles Cole for this one.)

[Editor's comment: The Alchemist has performed this upgrade on two BB's and heartily endorses it. The Alchemist also forgives Bob Puff for initially supplying the wrong pin connections to the LS139, which made the BB behave very weirdly and resulted in many colorful expletives. - BP]
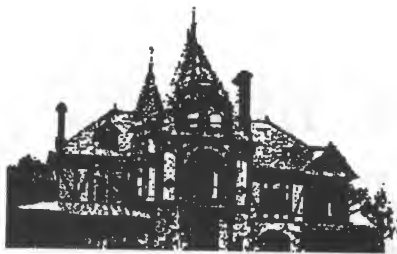
The Garret
The Last Word On 8-Bit DTP
Ed Hall, AC Staff Columnist

DTP Caveats

The first thing you should know about desktop publishing is that it's not a walk in the park. Don't expect the ease and simplicity of Print Shop. No matter which platform you're using, DTP is a lot of work: patience isn't just a virtue, it's a necessity.

The next thing to understand is that you need a sense of design. Remember, before DTP came along, preparation of camera-ready copy was done by graphic artists. Each page is a blank canvas on which illustrations, headlines, and blocks of text are positioned to create an overall pleasing effect. As well, the design must be functional: the reader's eye should move effortlessly from column to column.

For DTP, there's one clear advantage the more powerful machines have: greater screen resolution. This allows text and graphics to be compressed to a relatively small size on the screen and still be legible. DTP programs for the Classic Atari all use GRAPHICS 8 screens for the highest possible resolution, but even then it's not good enough to accurately represent more than about a quarter of a page. To get around this obstacle, most programs use four to eight screens to make up a single page. The variety of approaches has a marked effect on program design and performance.

Last time I discussed Newsroom and News Station. This time I want to look at some newer, and perhaps better, DTP programs.

What A Dump!

Page Layout and Hi-Resolution Dump are printing programs designed specifically for DTP. Written in TurboBasic, they combine eight picture files (62-sector, GRAPHICS-8 screens) to produce a single page. How you create those screens is up to you. You can use a GRAPHICS-8 art program (such as Graphic Master, Design Master, or R-Draw), or you can use the text and graphics editors of another DTP program.

Hi-Resolution Dump requires a 130XE or 800XL with expanded memory. It loads up to eight screens into memory, and allows them to be placed in any of eight positions (two columns of four each). Then it dumps everything to the printer in a single pass.

Though very similar, Page Layout has a few interesting differences. It prints faster than Hi-Res Dump (though the disk is constantly accessed during printing). It lacks a fancy user interface (all you see are a few prompts), but doesn't require the extra memory of a 130XE. Best of all, any screen can be used as a banner by expanding it across the full width of the page. To do this, you need to make a few simple changes to the code. A demo program shows you how.

Page Layout by John McGowan comes with full docs, including an interesting technical discussion on DTP. It can be found on the AIM disk for May 1993. *[Editor's note: these programs are all mostly PD or shareware. Vendors such as LAPD, C&T, MWPD, or ZTM typically feature them in their catalog listings. Check the Vendor's List in the October '93 AC for contact info. - BP]*

Digital Editor

Digital Editor is another TurboBasic program which (like Hi-Res Dump) originated in Germany. However, the current version (3.6) has been vastly improved by Tom D'Ambrosio. Unlike Hi-Res Dump and Page Layout, Digital Editor is a complete DTP package with separate text, graphics, and printer modules. A detailed tutorial is included.

The text editor allows the use of any standard Atari font in two sizes. It also has an 80-column capability, and can accept graphics in the form of clipart. Text can be typed in, or loaded from a file created with a wordprocessor. The latter is an important feature, and one which not all programs have. However, only 22 lines of text will be accepted; there isn't enough room on the screen for any more. This means a separate file is needed for every 22 lines of text.

Screens are saved in a non-standard format and must be converted to 62-sector files before they can be printed. The converter program also allows you to create your own clipart by defining and then saving portions of any 62-sector, GRAPHICS-8 screen.

The graphics editor is a good GRAPHICS-8 art program with features such as text, fill, mirror, and zoom. Curiously, it doesn't accept clipart. It does have a cut-and-paste function, and can use any standard Atari font.

Up to this point, Digital Editor performs well. But when we get to the crucial stage of printing, problems develop. The printer module is a modified version of Billboard, which requires several settings (print density, height, and width) for each printout. Unfortunately I was never able to find the right combination for the output I desired. Although this may well have been due to my own ineptitude, an even larger problem is that Billboard will print only one file at a time. Since at least four are needed for a page, you must roll back and reposition the paper for each pass. Too much hassle for my taste.

Fortunately, Digital Editor's modular approach makes for an easy solution. Just use Hi-Res Dump or Page Layout for your printing tasks. Tom D'Ambrosio recommends using four screens per page (with Billboard); my preference is for eight screens (when using Hi-Res Dump or Page Layout).

Alas, there remains one more problem, and it's a rather serious one. Digital Editor's text editor leaves a blank line at the top of every file—a big problem if you have a chunk of text that must be continued on the following screen. This is the same problem I experienced with Newsroom (discussed in the last issue). For this reason, I can't recommend Digital Editor for newsletters, which contain a lot of text. However, for shorter applications (like flyers), it should work just fine.

Page Editor

Page Editor is a complete DTP package with many interesting features. Like Digital Editor, it is a complete package utilizing the TurboBasic environment. However, its approach to DTP is somewhat different from other programs we've seen so far. It divides each page into four

horizontal sections, or panels. This means the width of each screen corresponds to the full width of the printed page, and requires using an 80-column text editor. If you have a RAMdisk, Page Editor can hold all four panels in memory at once, and each one can instantly be called up with the press of a key. This is a great feature.

Page Editor has some powerful text handling capabilities. The editor itself is serviceable enough; it supports inverse and graphics characters, has a block move function, and adjustable margins (useful for multi-column text entry). Any 9-sector font can be used, and its height and width adjusted.

But its best text-handling features reside in another module, the ASCII text converter. This allows you to create your text on a wordprocessor, then load it into Page Editor. You're given a choice of one, two , or three columns. The program will then *automatically* load your text into the designated panels in the correct order! I know of only one other program which attempts this feat: News Station Companion. It's a very useful, powerful feature. (Unfortunately, the program seems to have trouble with 3-column format. When I tried it, not all columns were the same width, even after adjusting the margin settings).

Page Editor's text and graphics editors are encompassed within the same module. A command switches you between modes, which is a lot handier than having to load in a completely new module, as with Newsroom or Digital Editor. The best feature of the graphics editor is its cut-and-paste function. You can specify a portion of a GRAPHICS-8 screen and save it for use as clipart, or move it, duplicate it, double its height, make a mirror image, or reverse it. Using another module, you can convert Print Shop icons into clipart for use with Page Editor.

The proof of a DTP program is in the printing, and with Page Editor there's a slight downside: the printed output is somewhat flattened. This is most noticeable when using Print Shop icons: they look rather squashed. Another problem is that you may not be able to get a proper-sized printout. I couldn't. If this happens to you, look for an explanation and fix by John McGowan (the same chap responsible for Page Layout) in *AIM*'s disk for May 1993.

There are a couple other minor problems: the Break key isn't disabled, and the program crashed more than once while I was using it. Nevertheless, it's a welcome addition to the ranks of Atari DTP. Written by Alan Kirk and in the public domain, it comes with full docs, a selection of clipart, and can be found on *AIM*'s disk for April 1993.

Daisy-Dot

Now you must forget all I've told you about DTP (except the parts about design and hard work), for here is a family of programs which take a completely different approach. Paradoxically, they produce printed text of unsurpassed quality.

Daisy-Dot 2 and Daisy-Dot 3 come from Roy Goldman, a talented programmer who last year won a $10,000 scholarship from Microsoft. While both programs produce gorgeous output, Daisy-Dot 2 is limited to relatively small font sizes (a good reason in itself for owning Daisy-Dot 3). The two programs also handle graphics differently; version 2 can incorporate 62-sector pictures into a print-out using the Verbatim command, while version 3 requires you to convert a picture into a font before it can be used.

A third program, Dot-Magic, is a clone of Daisy-Dot 2

with some useful additions: it has both a label and a typewriter function, and it is capable of using both MicroPainter and MicroIllustrator picture files.

So, what's the secret of these programs? Like Hi-Res Dump and Page Layout, they are specialized printer dumps with no text or graphic editors. A word processor is used to create a text file, in which are embedded special commands for Daisy-Dot to execute. It's a lot of work and, until you get good at it, you'll find yourself doing a fair amount of test-printing. Two-column printing is possible, but requires two passes through the printer-- make a goof on the second pass and you have to do the whole page over again.

Despite the amount of work involved, Daisy-Dot is the choice of many people doing 8-bit DTP. For example, there's Dark Star, the movie fanzine I reported on in my very first column. There's the bimonthly newsletter produced by the Tyne & Wear Atari User Group in England. And then there's Dave Richardson's 47-page user manual for Daisy-Dot 3. Not only is Dave's manual a very useful reference, it's also an example in itself of DTP on the Classic Atari. If you have Daisy-Dot 3, you need this manual! For further info, refer to Dave's column in the June '93 issue of *AC. [Editor's note: In the present issue Dave has pushed Daisy-Dot 3 to the limits of today's technology: it even works with laser printers! - BP]*

Xlent Printware

The very first Atari DTP programs were Typesetter and Page Designer, two titles in Xlent's Printware series. Both are complete DTP packages, but since they've been around for so long, I'll touch on them only briefly. Their approach is completely different from any other program. Like a spreadsheet, they both create a virtual page which is much larger than the screen itself. To see the entire surface of the page, you must scroll the window in the desired direction.

While Typesetter is capable of high-quality output, working with such a limited view of the page can be rather excruciating; for this reason, Xlent suggests using Page Designer (a much simpler program) to create a rough layout which can then be loaded into Typesetter and used as a sort of roadmap. Supporting material from Xlent includes two utilities, Rubber Stamp and P.S. Interface; and graphics, fonts, and layout grids from third parties.

A review of Typesetter, Page Designer and Rubber Stamp appeared in the February 1986 issue of *Antic*. In *ANALOG*, Typesetter was reviewed in issue 40, and Page Designer in issue 46. Particularly interesting is an article which appeared in the April 1992 newsletter of ACE of Syracuse, NY, and reprinted in the August 1993 issue of (FR)ANTIC, which is the newsletter of the Alamo Area Atari User Association (an 8-bit only group). Written by Vernon Smith, the article provides some useful information on using Typesetter for producing 3-column newsletters.

The Others

Antic Publisher is the most limited of the DTP programs for the classic Atari. It appeared as *Antic*'s disk bonus for the December 1987 issue. It has a crude editor which allows you to enter text, points and lines. Sixty-two sector, GRAPHICS-8 screens can be loaded, but not text files. Antic Publisher comes with a few well-designed fonts (including Hebrew), but these are in a non-standard format; if you want more, you'll design your own.

All in all, it's slow and laborious work, and suited for little more than providing captions to pictures. Antic Publisher is written in Atari Basic, but seems to work okay with TurboBasic.

Page Marshall is a commercial program which is especially suited for mixing text with schematics, musical scores, charts, etc. It's a British program and may not be available in North America. If you're interested, see the review in New Atari User issue 39.

The Works (according to a description in a PD catalog) is a DTP program which requires XL/XE machines with RAM upgrades. Since I don't have the necessary hardware, I've never checked it out. Could be interesting.

DTP Tips

Every Christmas I do a family newsletter on my Atari, and in the past I've put together a number of government publications using a Mac/PageMaker/LaserWriter combo. Emboldened by this vast experience, I thought I'd offer a few tips. Remember, though, *my* needs and preferences may be very different from yours.

1. Don't expect one program to do it all for you. Become acquainted with the strengths and limitations of each one, and select what you need for the task at hand. This applies not only to how they handle text and graphics, but also to how many screens are needed to create a page. For example, compare the final size of a 62-sector screen when printed out by the following programs:

Hi-Res Dump:	one-eighth page
Page Editor:	one-quarter page
Antic Publisher:	one-half page.

2. When putting together a newsletter, I usually use Atari's standard font for the body of my articles. Among 9-sector fonts, I've not found anything else as readable.

3. I almost never justify the right side of a column of text. This is because justification on our Ataris is usually achieved by opening up ugly gaps between words; this often causes the reader's eye to move vertically rather than horizontally. Better to leave the text ragged-right.

4. When using Print Shop icons, keep them as small as possible. The larger they are, the coarser they look. The same applies to fonts. If you need really large text, don't use an expanded 9-sector font; it'll look too crude. Rather, create your headline or banner using Print Shop's Screen Magic (thus using fonts intended to be large), then convert the image to a GRAPHICS-8 screen.

5. I never hesitate to employ old-fashioned paste-up techniques; that is, I often leave blank spots in the text, print it out, then glue in a picture afterwards. Since my final product is a single master copy, it never gets any farther than the nearest xerox machine. If unwanted lines show up along the edges of pasted copy, a bit of liquid white-out on the master solves the problem.

6. Add some color to your productions. Last year my kids used pencil crayons to brighten up our newsletter.

Cool Hardware From England
Compiled by Managing Editor Ben Poehland

One gets the impression our British colleagues want the benefits of upgraded hardware but disdain the fuss and bother of hacking and soldering. Although we Yanks have traditionally prided ourselves on a "can-do" philosophy of just rolling up the old shirtsleeves and diving into the belly of the beast, I rather suspect even the most rabid hacker would prefer to do things in a less messy way if the means were available. Those of you interested in the "kinder, gentler" approach to hardware upgrades might want to look into the following recent offerings from the other side of the Big Lake.

YORKY—Want a 256K in your 800XL without even opening up the case? Check out YORKY, a plug-in module for the XL that emulates ICD's RAMbo board. If you can't install this in less than a minute, call your 9-year-old to do it for you. YORKY is a sturdy grey box with two plugs and a socket. One plug goes into the power jack of the 800XL, while the socket accomodates the plug from your power supply. The other plug is a 50-pin edgecard connector dangling from a short length of ribbon cable; this attaches to the PBI port on your 800XL. That's it! Hands clean, no tools.

A well-written 8-page manual (originally printed on A4 but thoughtfully cut off to accomodate the American "quarto" paper size), and utility disk nicely round out the package. Cost is 55 pounds Sterling including shipping (about $80 U.S.). A lot more expensive than the Wizztronics board reviewed by Mike Jewison in the April '93 *AC*, but also about a zillion times easier! YORKY is available in limited quantities, get one while you can from: Richard Gore, 79 Sprotbrough Road, Sprotbrough, Doncaster, England DN5 8BW. Contact Richard on the Internet at: rg9@unix.york.ac.uk . It's been my experience that some British vendors will occasionally accept cash payments in US dollars, so you can avoid currency hassles. But, inquire first!

StereoPLUS—If you always wanted stereo sound but were terrified at the prospect of wrecking two POKEYs at once by soldering all those pins together a-la-GUMBY, check out the StereoPLUS upgrade board from Gralin International. It's a plug-in version of GUMBY with a few added bells and whistles. Okay, yeah, you *will* have to open your XL/XE machine to install it, and there are <cough!> two wires to solder, plus you'll have to drill a few holes in the case to mount the extra audio output jacks. But, anyone who's done a GUMBY upgrade will instantly recognize that's a whole lot less work than installing Chuck Steinman's original GUMBY hack. The StereoPLUS board plugs into your motherboard's POKEY socket. If you're unfortunate enough to have a soldered POKEY, all is not lost. Gralin thoughtfully provides a spare 40-pin socket, and the professionally printed 8-page manual gives instructions for removing the soldered POKEY. Installation instructions are accompanied by clear diagrams.

StereoPLUS comes in a clever professional-looking package that opens like a book, and it includes a software disk containing some TurboBASIC sound demos. As of this writing the final price hasn't yet been announced, and Colin Hunt of Gralin was doing a final sprucing-up of the docs and utility disk. Hopefully the product will be readily available by the time this issue of *AC* is out. Write to: Gralin International, 11 Shillito Road, Parkstone, Poole, Dorset, England BH12 2BN for information on price and availability of the StereoPLUS upgrade. Colin Hunt of Gralin has been known to sporadically haunt the Internet at: colin.w.hunt@sprintintl.sprint.com .

AC's 8-Bit Alchemist has been dying to rip into these goodies and spend some time prodding them to see what jumps out. Unfortunately there's a mean old managing editor here who won't let The Alchemist have any fun and is keeping the poor Alchemist in jail while he works on getting the affairs of this magazine in order. <sigh>

The Fitting Room
All Good Things...

Mike Jewison, AC Staff Columnist

It's A Small World

This being the final issue of *AC*, I'm feeling somewhat philosophical this month. Forgive me while I ramble a bit.

It's funny the way things work out. I live in a relatively small community about 40 kilometers north of Toronto, and to the best of my knowledge I'm the only owner of an 8-bit Atari in town. I've always felt a little isolated; even back in the heyday of the 8-bit computers all my friends owned Commie-64's. That's one of the great things about modems and networks: they really do turn the world into a global village. I may not have any 6502-based Atari friends in town, but via computer networks I have friends like Ben Poehland scattered over half a continent (actually, Ben's much more scattered than that, but that's another story :-). Considering the frequency and ease with which I correspond with my net-friends, they might as well live down the block. In fact, this magazine you're enjoying right now wouldn't have existed if it hadn't been for computer networks.

The staff of *Atari Classics*, too, is scattered around the world: from Itay Chamiel in Israel, to Don LeBow in Hawaii, to fellow Columnist Ed Hall who lives in Yellowknife, near the Arctic Circle in Canada's Northwest Territories. I've never met any of these people face-to-face, although I do communicate with some of them through electronic mail on an irregular basis. As fate would have it, my sister (who doesn't have the passion for computers that you and I do; I mean, she owns a Tandy of all things!) also lives in Yellowknife. So here you have Ed and me, thousands of miles apart, never met, both columnists for the Great Experiment known as Atari Classics. Wouldn't you know it, my sister works with Ed Hall's wife. Small world, indeed!

Back To Hardware Hacking

Last issue I made an ill-advised foray into the world of software, so I decided this month to wander back over to the hardware side of things. I figured after the problems of last time that hardware hacking would be easier on what's left of my nerves. Yeah, right.

As you know by now, my primary hacking computer is an 800XL which resides inside an old XT computer case. To the XL I have slowly added a number of peripherals: a TransKey adapter and 101-key PC keyboard (from Data-Que), 256K memory upgrade (from Best Electronics), and a Black Box/Floppy Board combination (from CSS) to which I've connected two 5.25-inch floppy drives (360K and 1.2MB). One of the major features of the Black Box (BB) is its SCSI port which will support any industry standard SCSI hard drive. Even before I bought my BB I dreamed of owning a hard drive for my Atari so, figuring no time like the present, I decided to make the plunge.

New SCSI drives are plentiful these days. Although IBM clones are migrating from SCSI to other standards like IDE, many of the major players in the workstation market (like Sun and Hewlett-Packard) fully support SCSI drives. Many large SCSI drives now available have capacities from 1.3GB (that's gigabytes) and up. For my little ol' Atari 800XL that seems a little like overkill: I mean, I don't expect there was anywhere near a total of 1.3GB of software written for the Atari since it was first released back in the late 1970s. Just to put things in perspective, 1.3GB corresponds to roughly 15,500 SS/SD floppy disks -- as if I'm gonna have the time to play with over 15,000 software titles before I die!

Since I decided I didn't need a hefty drive like that (nor the hefty price tag that accompanies it), I decided to aim for a drive in the more modest range of 40-80MB. I thought this would give me more than enough capacity for my purposes at a reasonably low price. The problem is that new SCSI drives in the 40MB range are scarce as hen's teeth and, even if you can find a drive that size, the price differential between a 40MB drive and a 200MB drive is negligible.

I didn't feel like shelling out megabucks (OK, OK, kilobucks) for a new 40MB SCSI drive, so I decided to pursue an alternate route. I had heard a number of horror stories about people buying used hard drives but decided that for my needs, that was the way to go.

USENET To The Rescue

Many of you have probably heard of USENET, an integral part of the much-touted (and, if you ask me, vastly overrated) Information Highway called the Internet. USENET is the Internet's equivalent of a BBS and contains literally thousands of newsgroups. Some of these groups are devoted to the buying and selling of goods, so I decided to post a little notice seeing if anyone had a small SCSI drive they wanted to sell.

I got a number of replies, the most promising one from a guy who lives about 80km from my place. Alas, the deal fell through when he (apparently) took an offramp from the Information Highway and was never heard from again. A second posting resulted in a few more replies, the most promising of which was from someone who lives in Montreal.

After some haggling, we agreed on a price of $100 for a 5.25-inch, 40MB Quantum SCSI drive. I sent off a cheque (just a note to our Editor: since I'm Canadian, the word is spelled "cheque", not "check") and received the drive about a week and a half later. *[Editor's comment: okay, you win. It's not my fault you guys up there can't spellllllll. - BP]* I immediately ripped open the package, lovingly fondled the drive (making sure I was suitably grounded) and waited impatiently for quitting time to arrive so that I could take it home and plug it into my BB.

Grinding Away

Of course, I couldn't just get home and head straight for the computer. Oh, no. I first of all had to help get supper ready for the kids, entertain the kids, and get the kids off to bed. So it wasn't until around 9:00PM that I was able to slip off into the basement... er, Fitting Room, with the Quantum.

I had scavenged some scrap 50-pin ribbon cable and a couple of IDC (crimp-on) connectors from our recycle pile at work to fabricate a cable to connect the hard drive to the SCSI port on my BB. I plugged everything together, connected one of the power connectors from the PC power supply to the Quantum, and fired everything up.

The first thing to be done with a new hard drive is a low-level format. CSS includes a floppy disk with every BB, and included among the programs on the floppy is a routine which performs a low-level format of the drive. I loaded the formatting routine-- and it promptly told me the hard drive wasn't responding and that I should check my cables, which I did. Everything was fine, at least with the cables, which meant the problem lay somewhere else.

The documentation for the BB states that any hard drive connected to it should have parity disabled. On the underside of my Quantum was a row of eight jumpers, one of which was undoubtedly for controlling the parity. But since I received no manual with my used hard drive, I was at a loss to figure out which was the proper one. Each of the jumpers was accompanied by a two-character code which, I figured, likely indicated the purpose of the jumper in question. Some of them were trivial to decipher: the three marked A0, A1, and A2 obviously allowed you to the set SCSI ID address of the drive (from 0-8). Some of them were less intuitive, like SS and WS. But there was one marked PE, which I quickly deduced to mean Parity Enable. (Actually, I wasn't really all that quick. And the route by which I finally arrived at this conclusion was extremely convoluted, so I won't bore you with the details here.) I removed the jumper, powered everything back up and, sure enough, the BB and the Quantum could talk.

[Editor's note: The newer Quantum "slimline" 3.5" *drives in the ELSxxS series don't even HAVE a parity* *jumper. These drives are sold with Parity permanently* *enabled, and there's no way to change it. If you get such a* *drive, contact CSS right away. For a small fee, CSS will* *provide you with a PAL chip you can add to your BB to* *overcome the problem. Other Quantum hard drives, such* *as those in the "PD-" and "LPS-" series, do have parity* *jumpers you can set (as of six months ago, anyway). The* *parity jumper on these drives is marked "EP". - BP]*

I reloaded the format routine and started it up. The format only takes a few seconds, but afterwards it performs a verification of the media to ensure there aren't any bad sectors on the drive. The program told me I had just over 160,000 sectors on my drive, and since the verification would take some time, I decided to let it run unattended and check on it periodically. Unfortunately, the verification pass would never complete. I tried running the program three or four times, and each time I'd get an error message reading "Controller not responding. Check your cables." during the verification. The error was transient in the sense that it never occurred at the same part of the drive; it was truly random. Well, I knew the problem wasn't with the cables since I couldn't have gotten as far as I did with a bad cable, so I started thinking that my new drive was perhaps a little flaky.

I sent off an email message to Bob Puff, proprietor of CSS, to ask his opinion. Bob told me the verification routines in the BB formatting program drive the mechanism extremely hard, harder than in normal usage. One suggestion he offered was to try formatting the drive but bypass the verification. Of course, the risk with this is that if the drive is flaky I would very likely lose data down the road if and when the drive failed. As an experiment, then, I fired up the computer to see if Bob's suggestion would bear fruit.

Scuzzy SCSI

I never got that far. Once the hard drive spun up it emanated the most horrendous grinding sounds I've ever heard. And when I tried to run the formatting software it told me it was doing something but the busy light on the drive stayed dark. I got this sinking feeling in the pit of my stomach: my hard drive had just died.

So I wrote back to the guy from whom I'd purchased the drive. He claimed it was in perfect working order when he shipped it (which I have no reason to doubt), but the fact remained that when received its most productive use was as a doorstop. I tried to negotiate some compromise deal with him but he just stopped responding to me email pleas. He had my money and I had a dead hard drive—and the Latin phrase "caveat emptor" running through my head.

Needless to say, I was pi..., uh, miffed. So much so that I spread news of my misfortune to anyone who came within earshot. (Actually, I was able to get a lot done at work that week because people started avoiding me like the plague.) One person to whom I told the story is a guy who runs his own company and, among other things, dabbles with bringing dead equipment back to life. He asked if I'd let him have a look at the drive. Since it was useless to me I gave it to him. About a week and a half later, he got back to me and told me the drive was shot. (At this point, I'm thinking "Really? I wouldn't have guessed!") He then offered to buy it from me and also threw in a working, 40MB, 3.5-inch used Quantum drive in a combo deal. Well, my parents didn't raise any stupid sons so after thinking about it for a few milliseconds I said "yes". In a somewhat dazed state, I left his place $20 lighter but with a working drive. (And if it didn't work, I knew where to find him!) I still don't know what he wanted with the dead drive but, to be honest, I'm not at all sure I want to know anyway.

To avoid making a long story downright tedious, I took the new drive home, plugged it in, booted the computer, and ran the format software. The program came back and told me I had 16xxxx sectors on the disk. I then nervously watched the verification routine do its thing, and it finished with no reported errors. Hallelujah! I finally had a working hard drive!

Once the low-level format is finished, the drive is ready to use. The next thing to do is decide how to partition the drive. Partitioning allows you to segment your drive into a number of so-called "logical" disks-- "logical" because although they aren't individual physical drives you can consider them so. I brought up the BB Configuration Menu and set up the equivalent of eight 720-sector partitions on the Quantum. I picked 720 sectors because that matches the number of sectors found on a single-sided floppy. My intention was to copy a number of boot floppy disks to these partitions and then configure the BB to load one of these partitions as the computer booted.

As an experiment, I copied my backup copy of Scarborough Systems' "Your Personal Net Worth" to a 720-sector partition, set that partition to be Drive 1, and saved my drive configuration to the hard drive. I then power cycled the computer, and upon rebooting YPNW loaded in from the hard drive. (The computer's actually pretty stupid in all this. All it knows is that it wants to boot the floppy in Drive 1. But the BB intercepts the computer's request to access Drive 1 and reroutes it the the hard drive partition instead.) It takes YPNW about 3 seconds to load in from the hard drive- roughly three times faster than it takes from an UltraSpeed formatted floppy and about 10 times faster than from the original disk. I copied a bunch of other programs I have to other partitions and tried them out as well. The results were generally similar to those of YPNW: programs would load from the hard drive roughly 10 times faster than from floppy.

I'm still playing with the hard drive. Eventually, I plan on setting up a number of large partitions (several megabytes each) and format them for use with MyDOS and SpartaDOS. There are also a number of great shareware packages out there which would benefit from their own large partitions: things like TextPro and DaisyDot III. Alas, as with all things around our house, time is always at a premium.

The moral of this month's adventure? Never buy a used hard drive from someone unless you can go and pick it up yourself. Okay, maybe I'm overreacting, but I don't ever want to go through that frustration again.

Into the Sunset...

There's always a tinge of sadness when a chapter in one's life comes to a close. I'm saddened by the demise of *AC*, particularly since I feel like one of the half-dozen parents who saw it through gestation two years ago. I'm also saddened by what's happened with the magazine's readership and potential readership. I'm aware of a number of people who consider themselves avid 8-bitters but never subscribed, giving excuses like "Gee, I always meant to get to it" or "I don't have the money right now, maybe in a couple of months".

Unfortunately, *AC* couldn't wait for them. I guess these people figured *AC* would always be there, waiting with open arms for new subscribers. The truth of the matter is that as the number of subscribers began to dwindle, *AC* passed below the critical mass necessary for self-sufficiency. I guess what disturbs me most about all this is that this is not just another commercial publication disappearing due to lack of revenue. *AC* has always been "by the user, for the user". What we have here, then, is the 8-bit commmunity failing to support itself. Perhaps even some of the remaining hard-core users are being swayed by newer technology. Maybe people just plain didn't like *AC*. Whatever the reason, it appears our ranks are diminishing. I, for one, will be here to the bitter end, but it's starting to get a little lonely.

Although we've seen the last of *Atari Classics* in its present incarnation, it's entirely possible we may *not* have seen the last of The Fitting Room. Keep your eyes open—you never know when or where I may pop up again.

And in the words of some amateur astronomers when they part company, may you have clear skies ahead.

The Starting Line

by John Hardie, AC Resource Editor

Some Basic Stuff

This month we'll take a more in-depth look at the different models of classic Ataris out there and discuss their features. But first, let's review some basic concepts about computers in general. I'll start with memory.

A computer's memory is measured in bytes. A byte is equal to one character whether it's a letter, number, punctuation mark, or even an empty space. The word "byte" is four bytes long, one byte for each character. When you say that a computer has 64K memory, you're saying it has 64 Kilo-bytes or 64,000 bytes of memory (remember kilo (K) = 1,000). A computer with 2 megs would have 2 million bytes of memory. This internal memory is called "Random Access Memory" or "RAM"-- don't confuse it with the storage capacity of a disk drive. A disk drive can hold large amounts of program data but can't execute or do anything with it other than to store or retrieve it. A RAM chip can hold the information in it until it loses power (you shut off the computer), or you put new information into it. Another kind of memory is called "Read Only Memory" or "ROM". A ROM chip stores information permanently (when it's manufactured), and thereafter the stored information can be read from the chip over and over. ROM memory is retained even after the power is turned off. A typical example of a ROM chip is a cartridge-based game you plug into your computer, or the operating system chip in the computer itself. If ROM and RAM seem a little confusing, don't worry; you'll get the hang of it after a while. Now that you know a few buzz-words, let's talk computers!

A Million Models?

Not quite, but eight different models of Atari 8-bit computers were released over the years. Let's start at the beginning with the Atari 400:

Atari 400: light brown case with a flip-top lid and membrane keyboard. The power switch, power jack, and I/O connectors are on the right hand side of the unit. The front of the unit has four joystick ports and the back has only the channel 2/3 switch and a built-in RF cable. The computer has 16K of RAM.

One of the main drawbacks of the 400 is its membrane keyboard which is just plain difficult to type on. The small smount of memory prevents it from being used for any serious applications. The machine doesn't have BASIC programming language built-in (not a big deal if you're just interested in playing games). It does have 4 joystick ports in front which means up to 8 people can play the game "Breakout" together. Some companies designed products for the 400 and 800 that used these ports specifically (a drawback for the later machines when ports 3 and 4 were removed). There were also hardware upgrades and modifications for the 400 that brought it up to par with the 800. At least one company designed a regular keyboard as well as upgrading the memory to 48K and adding a monitor output to the machine. However, in its stock form it's best suited for early Atari cartridge games.

The Atari 800

Atari 800: light brown case with a flip-top lid and a regular keyboard. The power switch, power jack, monitor jack, channel 2/3 switch and I/O connectors are on the right hand side of the unit. The front of the unit has four joystick ports and the back has only the built-in RF cable. The computer has 48K of RAM.

Although similar to the 400, the 800 computer featured two cartridge slots under the lid. However, with the exception of one or two specialized cartridges, the right slot was never used. The flip-top lid itself was removable to reveal four boards (one ROM and three RAM). This access to the ROM and RAM allowed companies to come up with quite a few different RAM upgrades and modifications for the 800. The addition of a monitor jack enabled the 800 to output a better picture using a composite monitor versus a regular TV. The rugged construction of the 800 is legendary: if you want a computer

that feels like a tank and is built to withstand a nuclear blast, the 800 is for you.

The 1200XL

Atari 1200XL: white and chocolate brown case with a row of silver keys and three LEDs across the top of the keyboard. The power switch, cartridge slot and two joystick ports are on the left hand side of the unit. The back of the computer has the power jack, monitor jack, RF jack, channel 2/3 switch and I/O port. The computer has 64K of RAM.

Many will argue that the keyboard on the 1200XL is the best one ever built. In addition to the START, SELECT, OPTION, RESET, and HELP keys, the 1200XL also featured four function keys, F1-F4 which controlled various options such as disabling the keyboard or enabling the international character set. The keyboard also has three LEDs labelled POWER, L1 and L2. The L1 and L2 lights were software controllable but were mainly used as part of the built-in diagnostic routine. The programmable HELP key was to allow software companies to provide "one-touch" help screens in their software, although only a few ever used it. The 1200XL only had two joystick ports (a sore spot among multi-player game fans). Internally, as I mentioned before, the 1200 has a built-in self-test routine that checks the computer's ROM, RAM, audio, video and keyboard functions. Also, the 1200 has a current-limiting resistor on the I/O port which basically prevents anything that draws power from the I/O port from functioning (like the XM301 modem). Another sore spot for many owners is that in redesigning the Operating System, some of the older 400/800 software was rendered incompatible with the 1200XL (a very short-lived model).

The New XL Computers

Atari went back to the drawing board and soon released the 600XL and 800XL:

Atari 600XL: white and chocolate brown case with a row of silver keys on the right hand side of the keyboard. The power switch, power jack, RF jack, channel 2/3 switch, parallel bus and I/O port are on the back of the unit. The joystick ports are on the right hand side of the unit and the cartridge slot is on the top above the keyboard. The computer has the BASIC programming language built-in and 16K of RAM.

Atari eliminated the four function keys and two LEDs from this and all future models. The (still only two) joystick ports were moved to the right side of the computer. The 600XL has no monitor output and can only be used with a TV (unless you upgrade the video output as described in the December '93 *AC*). The addition of the parallel bus, or PBI, enabled the user to upgrade the system's memory or add on other future components. The Atari 1064 memory module plugged into the bus and gave the 600XL 64K of memory. Internally the Operating System was slightly more compatible but still required the use of a "translator" disk for older software. Turning the computer on brings up the BASIC language, while holding down the OPTION key when doing so disables the internal BASIC and brings you up in the self-test routines of the machine's Operating System, or OS. Both BASIC and the OS are ROM chips installed on the computer's motherboard.

The 800XL

Atari 800XL: white and chocolate brown case with a row of silver keys on the right hand side of the keyboard. The power switch, power jack, RF jack, monitor jack, channel 2/3 switch, parallel bus and I/O port are on the back of the unit. The joystick ports are on the right hand side of the unit and the cartridge slot is on the top above the keyboard. The computer has the BASIC programming language built-in and 64K of RAM.

There isn't much difference between the 600XL and 800XL other than the physical size of the computer, the amount of memory and the addition of the monitor jack for output to a composite monitor. The 600XL and 800XL both used the same keyboard, which was a planar design. The 800XL was a big seller for several years, even after the XE series machines appeared. This is the most commonly encountered classic-8 machine at garage sales and flea markets. Many earlier units had socketed chips, which has made them valu-

able for hardware upgrades. Definitely the most popular of the XL series, maybe more numerous than any other Atari 8-bit model.

The XE Machines

Atari's final entries into the 8-bit computer market were basically updated XL models called the 65XE, 130XE and XE Game System (XEGS). For you trivia buffs, the XE stands for "XL-Extended":

Atari 65XE: grey case with START, SELECT, OPTION, etc. keys across the top of the keyboard. The power switch, power jack, RF jack, monitor jack, channel 2/3 switch, cartridge slot and I/O port are on the back of the unit while the joystick ports are on the right hand side. The computer has the BASIC programming language built-in and 64K of RAM.

The 65XE computer is really just a redesigned 800XL in a new style case. Atari dropped the parallel bus from the machine and positioned the cartridge slot in the back. The internal circuitry was changed, but the machine functions much the same as an 800XL. The 65XE and its big brother, the 130XE, are both the same size, and both are slightly smaller and lighter than the 800XL. The XE keyboard is quite different than the XL keyboard, being a sculpted design. Some users feel the XE machines generally have a mushy keyboard and a more flimsy feel, but this is a matter of taste.

The 130XE

Atari 130XE: grey case with START, SELECT, OPTION, etc. keys across the top of the keyboard. The power switch, power jack, RF jack, monitor jack, channel 2/3 switch, cartridge slot, ECI and I/O port are on the back of the unit while the joystick ports are on the right hand side. The computer has the BASIC programming language built-in and 128K of RAM.

Extra memory is the highlight of this model. Essentially a 65XE with enhanced features, the 130XE has an additional 64K of RAM that can be used in programs designed to take advantage of it. The additional memory can also be used to simulate a disk drive (a RAM-disk). Another feature is the addition of the Enhanced Cartridge Interface (ECI) which is functionally similar to and software compatible with the parallel bus on the 600XL and 800XL.

And Finally... A Game Machine???

Atari XE Game System: grey case with pastel colored START, SELECT, OPTION, etc. keys on the main unit. Detachable keyboard. The power switch, power jack, RF jack, composite audio and video jacks, channel 2/3 switch, and I/O port are on the back of the unit. The cartridge slot is on top of the machine while the joystick ports are on the side. The computer has the BASIC programming language and Missile Command game built-in and 64K of RAM.

Atari chose a new approach with their final entry into the 8-bit market by calling it a game system. The idea was that customers could go at their own pace with the machine. While great for playing games, the user would discover they already had a computer if they decided to upgrade. Pretty much a 65XE in a redesigned case, the XEGS also boasted the game "Missile Command" built-in, as well as the BASIC language. The unit was packaged with a light gun as well as the "Bug Hunt" and "Flight Simulator" cartridges. Unfortunately for Atari, the system never took off (pun intended).

Time To Move On

So there you have it: the major models of Atari 8-bit computers in a nutshell. There were a few others, rarities which never made it to mass production. In a future column we'll take a closer look at Atari-made and Atari-compatible disk drives, but that will depend on whether this magazine survives. As I said before, if I can be of any help in answering any questions, or if you have any comments, please feel free to contact me. You can write or call, or e-mail me on the GEnie online service. If calling please remember that I'm on Eastern time. Here's the info...

John Hardie
90 Judith Lane
Valley Stream, NY 11580
516-568-9768
GEnie: J.HARDIE
Hope to see you next time!

What Now?

Jim Hood

Since you're reading this, you know Ben Poehland decided to call it quits while it was still possible to make some refunds to *AC* subscribers who backed his efforts with their money. When Bob Woolley found that Ben was considering giving up the *AC* adventure, he started trying to work out an arrangement to keep the magazine going.

He asked if I would be interested in editing and laying out the magazine. "Sure," I said, not being noted for my highly rational behavior.

I hoped Ben would continue as Managing Editor with me doing the physical layout of the magazine and another of Bob's friends; also named Bob Woolley; doing the printing.

By then Ben was too worn down to continue on what looked to be an unpromising path, so we (the California crowd) talked it over and decided we could still pull off publication with the lower subscription level that Ben is hobbled with.

Now Ben, like you, doesn't know us from Adam, except for his work with Bob Woolley (the 8-bit one). He wants to honor his financial responsibilities to his subscribers and printer, Unicorn Publications, so he feels it is best to end his version of *AC* and let us do our thing as a fresh venture.

We new folk feel a sense of continuity is important. We want to continue the *Atari Classics* name and hope the current editorial staff will stay with us. We want to continue publishing bimonthly, with a disk available three times a year. We will continue the current subscription rates, and will even offer you a discount if you're brave enough to subscribe for two or three years.

So, if you're interested in this thing continuing, cash Ben's refund check and start a fresh subscription with us.

Yeah, we realize it's a big hassle for you. We hope you realize it's also a big hassle for Ben and us.

As I said, we hope the current editorial staff will continue with us. Including an occasional column from the 8-Bit Alchemist.

We are also trying to arrange coverage of the 8-bit BBS scene and hope to include articles for new users.

We haven't contacted any of the "old" staff yet, but to give you a feel for what we'll be doing, Ben has given us this four page insert and we've imposed on a couple of our developer friends to provide you with some thoughts about their plans for the Classic 8-bit market.

At the Sacramento Atari Computer Exposition Bob Woolley had a chance to talk with Darek Mihocka about Darek's new PC Xformer. We give you our impressions in a third article.

Here we go...Coming?

Genlock for the 8-bit

An interview with Michael St. Pierre, owner of MYTEK, an 8-bit Enhancement Company.

This is one of those phony interview articles, where a magazine gets information from someone and adds questions afterward. What next? Doctored photos?

What got you interested in doing an 8-bit genlock?

Genlocking the 8-bit Atari became kinda a pet peeve with me about 2 years ago. I guess I've never liked people telling me something is flat-out impossible when my gut feeling says otherwise. Seems like most of the 8-bitters still in existence today, must share this feeling, defending their right to choose in a world ever more dominated by IBM's. Well Atari people had gotten their 8-bit machines to do just about everything imaginable, but genlocking was sadly lacking. Having worked in the video field for several years, I decided to make genlocking my offering to the Atari spirit.

So Prism Studio has been in the works a couple of years?

Like I said, two years ago it all began. An earlier attempt came in the form of a luminance only genlock board, which was written up in the SLCC *Journal* Nov '91 and offered to the Public Domain. As first attempts go, it wasn't bad, sure it had it's share of problems and limitations, but it did indeed allow 8-bit graphics to overlay external video.

What were some of the problems?

The main problem with this early design, was it's inability to handle disk drive access without locking up the entire computer in the process. It also lacked color, being a black and white only form of overlay. Hey, it was genlocking after all, and on an 8-bit Atari none the less!

Evidently, you still weren't satisfied.

Well, another year went by, I'd learned more about color subcarrier regeneration, phase lock loops, VCO's, and a bunch of other techno stuff that all seemed to point to a second generation genlock device. I worked on these ideas whenever I could, but unfortunately I was unable to get any real lab time on the project for some six months to come. You see I was living in an RV, based some 30 miles from civilization, up in the mountains. My only source of power was a 100 watt solar array charging a battery, with a small inverter suppling the AC I needed for short term uses. This had been my great escape from the nuisances that life can sometimes bring (which happens all too often I'm afraid). Don't get me wrong, it was actually quite refreshing in many ways, kinda like a long vacation can be. All things pass, and a girl by the name of Jennifer brought me out of my RV, and back to the city with a real house. Of course, this meant that I now had the means to do some serious hardware development, thanks to PG&E and an understanding girlfriend.

Behind every great man stands a woman and giant utility company.

It was still to be a grueling eight months to come, before I had a working color genlock, and as the time went by, I often found myself wanting to just give up. You see, it

quickly became apparent that designs I had worked out in my head, and that had seemed so simple, were just not right. Like everything else that's worth anything, it required many cases of cola, some hair loss, and many dead brain cells to finally resolve all the problems. It also required several cardboard boxes to hold the carcasses of early design prototypes, but out of it all emerged what I have come to call PRISM STUDIO.

Why PRISM STUDIO?

Well, it just seemed appropriate, PRISM representing color, and STUDIO representing it's use as an editing, creative, studio-like tool. I guess the thing that seems most studio-like about it, is the TITLE FADER, which allows for dissolves.

And this time you decided to supply a finished product, rather than schematics?

Along with this new genlock product, has emerged a new company, MYTEK. I felt that this was such a significant accomplishment in the genlocking arena, that PRISM STUDIO needed to become a real product, available to anyone in the Atari 8-bit community. Establishing a company to produce this product was the only way to go, it's just too complex to be offered in kit form.

Good luck with this new offering.

With PRISM STUDIO as our launching product, MYTEK hopes to do well, and provide additional enhancements and/or products in the years to come.

Impressions:
Sacramento Atari Computer Exposition '94
Jim Hood

The second Sacramento Atari Computer Exposition was held on March 12 and 13 at the Towe Ford Museum in Sacramento, California.

The event is sponsored by STAR, the Sacramento 16/32-bit Atari user group, and focuses on Atari TOS based computers.

There has been good support for the fair from Atari and the western portion of the 16/32-bit development community both years.

Last year Atari showcased the Falcon030; this year the Jaguar.

They donated both a Jaguar and Falcon030 for this year's raffle. There were also hourly door prizes donated by Atari, the developers and the participating retail dealers.

All the northern California clubs that support 16/32-bit Atari computers appeared to have tables at the fair. Bob Woolley, Peter Chen and I represented the San Leandro Computer Club.

Bob decided to try flushing out any 8-bit users at the fair by demonstrating his 1200XL with the built in IDE hard drive. He set out the 1200XL with the top casing removed, letting people see the hard drive as graphics flashed from it to a monitor.

It was an effective mouse trap. People who had no interest in 8-bits didn't understand the significance of the hard drive and went on about their business. A lot of those that did understand asked about it. Bob met quite a few Sacramento area 8-bitters that way.

People attending the fair seemed enthusiastic and we enjoyed having a chance to talk with many of them.

We hope the STAR user group found their work rewarding enough to do it again next year.

It was a rewarding experience for Peter Chen. Mike Fulton, from Atari, drew Peter's name in the Falcon030 raffle.

The president of STAR won the Jaguar. A committee is being formed to appoint a special prosecutor to look into allegations of wrongdoing.

Darek Mihocka, of Branch Always Software, was in attendance. He was selling his Gemulator board for running ST software on PC clones and demonstrating his PC Xformer software. He is still polishing up the PC Xformer package, improving some of the existing routines and adding a couple of others to expand compatability.

Bob Woolley had been luke warm to the 8-bit emulator concept before the fair, but after some lengthly practice using the Xformer while talking with Darek, he is sold. The only thing that really disappointed him was Darek's refusal to sell him a copy of the existing version of the software.

The Xformer looks like it will be a good, clean piece of software that will run most 8-bit programs. Overly copy protected programs and those that break too many rules might make it stumble, but the sampling that we saw was impressive. Speed is good and text looks great on an SVGA monitor.

Introducing Fine Tooned Engineering

Mike Hohman

The rumor of someone buying ICD's 8-bit product line, is true. Rights and existing stocks were acquired by Fine Tooned Engineering of Scotts Valley, California. Owner, Mike Hohman, gives some insight into his plans.

Hi, my name is Mike Hohman, from Fine Tooned Engineering. Even though I was asked to write this article, it's something that I've always wanted to do, but simply procrastinated doing for "Whatever the Reason" might be.

I consider myself an Atari Enthusiast, and will continue to do so for the rest of my years. I guess you could call it being past the point of no return!

I wanted to touch base on a few issues, along with giving an update on what's Really Happening with FTe and the ICD/OSS products.

The first topic is "Fear of Failure." I think that this is an important subject and applies to all of us...regardless. If I had listened to 99.99% of my peers about the current state of the Atari 8-bit Community, the ICD/OSS rights would have never been acquired by FTe.

I think that the transfer occurred as the result of a lifetime of frustration with Waiting for Something to Happen. It's been a really great opportunity to confront my fear of failure directly, in addition to putting FTe in the driver's seat to Get Something Done!

I never subscribed to *Antic* or *Analog*, even though I have virtually every issue of both magazines. I never subscribed to *Current Notes* or *Atari Classics* either.

Most of the Atari software and hardware that I currently own wasn't purchased directly from dis-

tributors or manufacturers, because I could get a "better deal" buying things second hand.

Although I've finally managed to assemble a really nice Atari setup, the sad fact remains that there is virtually no new commercially available hardware or software. Kind of self defeating, isn't it?

I remember how I felt both times when *Compute* and *Computer Shopper* stopped Atari support.

Then *Analog* folded, followed by *Antic*. It was only after they were no longer available that it occured to me that I would NEVER be able to

write to Readers Feedback or share an article or program with all of you.

Do you think I'd learned my lesson yet? Nope. I never submitted an article to *Atari Classics* either! Talk about a Procrastination/Fear of Failure double whammy!!

Well, it's to the point that *Atari Classics* is changing it's current form and almost went the way of *Antic* and *Analog*. This was the last straw for me, and the article that you are reading is proof that "Mike has final-

ly gotten off his butt" and is doing something! The funny thing though is that I *have* been doing something. For the past 3 years. It's just that only a handful of people have known about it!

That something is a really neat project called the MARS 8, and was simply delayed because I needed the ICD/OSS rights to finish it. It's about 85% done, and 10% of what remains is legal paperwork, with the other 5% being the polishing touches. I can't say much more about the project as I *write* this article, but I probably can by the time you *read* it. So I encourage everyone to call, or write for info...

> Fine Tooned Engineering
> P.O. Box 66109
> Scotts Valley, CA 95067
> (408)Get-REAL (Info/Help) or
> (800)For-6502 (Orders only)

I really hope that everyone resubscribes to the new *AC*, brushes up on programming and/or writing and supports this last ditch effort!

Things are actually brighter now than they ever have been, it's just simply a matter of opening your minds.

In addition to MARS 8 and the existing products we acquired from ICD there are about 5 new projects in the pipeline, and with them all being done concurrently, it will take a few months to get caught up.

When our first product ships, there will be several more hot on it's heels!

Moonlight Workshop

Printing with the Epson Inkjet

Jeff McWilliams, AC Staff Columnist

I'm Back!

It's a shame that after missing so many issues, the reappearance of "Moonlight Workshop" coincides with the last *Atari Classics* we might ever see. It's wasn't by design that I've been conspicuously absent from the magazine I helped create. When I moved from Houghton to Alpena (in Michigan) in August of last year, I had hoped to quickly acclimate to my new surroundings and pick up the expanded memory series where I had left off. Those intentions proved overoptimistic.

The Epson Inkjet

In September I upgraded from a 24-pin printer to the new Epson Stylus 800 inkjet printer. For the first four months I was much annoyed with "teething problems". First I had to find a brand of inkjet paper I could rely on. Weyerhauser First Choice Premium Laser Paper turned out best. At $5.79 per 500 sheets plus $1.50 S&H from MEI/Micro, it's cheaper to buy this good stuff by mailorder than to buy the cheap-o copy machine paper sold locally.

Two months after I purchased the unit it suffered a mechanical failure and ceased depositing ink on the paper. It worked okay when I got it back from the local Epson service center. Then it started depositing too much ink on the paper, producing smeary letters and random lines of stray ink. Another visit to the dealer, and finally the printer worked properly.

I then had to worry about purchasing ink cartridges. The Stylus 800 ink cartridge runs $20, which I thought a bit pricey. (By comparison, the HP Deskjet 500 cartridges from MEI/Micro cost $25.37.) It's amazing how fast you use up twenty bucks' worth of ink, even with the printer in economy mode. Unlike dot matrix printers, there don't seem to be generic versions of ink cartridges, just refills-- if you can find a place that sells Stylus 800 refills. After some searching, I discovered V-Tech in Hatfield, Pennsylvania. They sell refill kits for $12.50 that can refill a cartridge up to eight times. It took me two months and two phone calls to V-Tech before I reduced the refilling procedure to a more or less error free science.

I also spent several weeks and a couple hundred sheets of paper reconfiguring every wordprocessing and database application I use. Since I use Textpro 5.0X, AtariWriter-80, and Turboword/Turbofile pretty much equally, I spent triple the amount of time any normal, sane person would to get my computer system accustomed to its new printer environment. Three printer utilities written in Action!, five macros written with TextPro, and one homemade high-speed printer interface later things are beginning to feel like normal. (Whatever "normal" is for a diehard Classic Atari user these days!)

The printer as a whole is a fine product. A friend who works in a local print shop declared its output to be most impressive. I especially appreciate how quiet it is compared to the whining of my old dot-matrix printer. Not having to deal with fanfold paper is also a plus. I attribute the mechanical failure to my usual bad luck: I got the one unit in a thousand doomed to failure. It happened before with my first 24-pin printer, my first memory upgrade, my first color monitor, and my first floppy drive. *[Alchemist's comment: Gee, and all this time I thought it was ME the Murphy demons were after! Well, now I don't feel so persecuted.]*

I've come to expect the worst when I purchase hardware these days. I guess that's why I was so delighted when I installed the Ultraspeed+ OS and the XF551 single drive upgrade from CSS into my hacked 1200XL. Instead of sputtering and dying like everything else I buy, these two products actually worked, and nicely too. I heartily recommend the US+ OS to *any* XL/XE user, and CSS's line of XF551 upgrades are a no-lose situation for all XF551 owners out there.

Doom For Dot Matrix?

Dot matrix printers are slowly marching off to obsolescence. Yeesh, I feel like a *PC Magazine* columnist saying that. Those guys are always preaching about how everyone needs a Pentium-class PC because anything slower or less sophisticated is worthless. They think everybody needs mainframe power to do even mundane household tasks like manage the Christmas mailing list and balance the checkbook.

In the case of the dot matrix printer though, I think it's true. Faced with the quieter operation, improved speed, sharper output, and declining prices of inkjet and laser printers, the trusty old pin-fired models are being overlooked in favor of the competing inkjet/laser technologies. More and more dot matrix printers should start appearing at garage sales, auctions, and trash dumpsters, ripe for the picking by bargain-hunting 8-bitters.

It also means that a time will come when some percentage of the 8-bit user community will have an inkjet or laser connected to their Atari. Maybe it's because the 8-bit is sharing the printer with a PC; or perhaps it's just that guys like Frank Kweder [*AC*, August '93] and I can't resist the idea of seeing our old Atari driving the newest, fastest gizmo from the mainstream computer market.

Paper & Labels for Inkjets/Lasers

Whatever the reason for purchasing the printer, woe to the unprepared user who expects it to perform like the old pin-fired models. Unlike dot matrix printers which use continous fanfold paper, inkjets use single sheets of 8-1/2x11 or European A4 paper. Due to the design of the

paper handling mechanism, most inkjets can't print at the very top or bottom of the page. This alone will persuade you to get reacquainted with your user manuals for several hours as you try to reconfigure the page characteristics. I had to change settings for total lines per page, printable lines per page, top margin, and bottom margin. The same setting doesn't work for every wordprocessor, which forces you to do a series of test prints followed by readjustments until wordprocessor and printer work together in harmony.

Any trouble you experience with a wordprocessor is trivial compared to the difficulty you'll encounter trying to print labels from a database or specialized label management application. Like paper, inkjet or laser printer labels come in sheets, not continous strips. And like paper, you can't print at the very top or bottom of a sheet of labels.

Because of this, the current crop of database and label printing applications for the Classic Atari are incapable of printing address labels on an inkjet or laser printer. Using a proportionally spaced font to squeeze in those long addresses compounds the problem. Not only does the application need to manage the top and bottom margins, it also has to send horizontal tab commands between each label to ensure proportional text lands in the right spot.

December Dilemma

In December I had to perform my annual ritual of mailing Christmas cards and developed a royal case of frustration/anxiety. If my new printer couldn't manage a fairly simple task such as a Christmas mailing list, then my Atari was becoming truly useless. The consequences of such a possibility were demoralizng. After years of continually upgrading my software and hardware, my efforts to keep my 8-bit capable of meeting my evolving needs would be nullified in one fell swoop by the purchase of a new printer! The very idea really got me down.

For the first time in my experience as an Atari 8-bit user, the software market failed to supply an application that embraced a new, economically available technology compatable with the Atari. When I needed a database for the Mail Campaign, Turbofile was there, utilizing the capabilities of the XEP80 and SpartaDOS's support of mass storage and subdirectories. Daisy Dot III pushed the envelope when it came to 9-pin dot matrix printed output, and TextPro's macro capabilities are nothing short of brilliant.

Now comes this inkjet printer technology. The same people who purchased hard drives and products like ICD's MIO or the Black Box from CSS could be expected to also upgrade to an inkjet printer. So here I am, printer in hand, and lacking a shrink-wrapped program to use with it. Our declining software market is increasingly becoming a chain that hobbles the Atari 8-bit way of life.

Lacking any readily available label software, I chose a simpler, quicker approach to solve my Christmas mailing problem. Taking Turbofile as a starting point, I modified its output with a Textro 5.0X macro. Once modified with the macro and printed as a TextPro document, the resulting output is rendered free of obstacles. Using a sheet of three labels across and ten down, I successfully printed proportionally spaced addresses with no problems. The macro automatically uses proportionally spaced san serif text and allows five lines of text per label.

The TextPro macro, Turbofile account setup and print format files, and several other TextPro support files are included along with TextPro 5.0x on Side B of the February '94 *AC* Software Disk. I designed the system to work with SpartaDOS-X. The main portion of these instructions covers using my address labels on a similarly configured system. I'll also give additional instructions for users who don't have SDX or Turbofile.

Using The Macro

Before we can print a set of addresses on labels, we first must have a database of addresses. This is done with Turbofile. Make sure XMAS.ACT and XMASLABL.FMT are on a Turbofile work disk, or in your Turbofile subdirectory. Initialize the new account with the "I" option from the Turbofile menu; the name of the account is "XMAS.ACT". Entering records into the database will now be fairly straightforward. The Turbofile configuration file is set up to accept six lines of text per record. I suggest only using five of these because the sixth line may be printed too close to the bottom of the label.

When you're ready to print labels, choose the Print option from the main Turbofile menu, select XMASLABL at the following screen, and do a CTRL-DOWN-ARROW at the screen after that. Then press "F" for file, and type in a filename, WITHOUT a drive specification. It will ask you which drive to save the file to, and whether you want the output in a single column format. Respond with "N" to the last question. The XMASLABL.FMT file does all the dirty work.

When Turbofile is done "printing" the data to the output file, go back to the main menu, choose "D" for disk operations, and "D" to exit to DOS. Set up your "SYSTEM" and "Textpro" disks if you have to, and make sure the Turbofile output file is in your current working directory. Be sure all TextPro-related batch files, TEXTPRO.COM, and TextPro's MACRO and configuration files (the ones distributed on the program disk as part of the label macro) are contained in either the main directory, your current working directory, or subdirectories listed in your SpartaDOS-X search path.

If you're not sure about what your SpartaDOS-X search path is or how to change it, grit your teeth and look it up in that dreadful SDX manual provided by ICD. See pages 4-47, 5-4, and 4-59. You can place a "SET PATH= ....." command in either your CONFIG.SYS file or AUTO-EXEC.BAT file, or type it directly from the command prompt.

Once you have everything configured and set up properly, all you have to do is type "-XP FILENAME.EXT [RETURN]" substituting the name of the file you gave at the print menu of Turbofile for "FILENAME.EXT". TextPro and the TextPro macros should do the rest. Be sure your inkjet/laser printer is online, and that labels are loaded. When Textpro finishes rearranging the label data, it will automatically start printing the labels. TFLABEL.MAX will exit to DOS when it is finished printing. If you are in XEP80 80-column mode you *don't* need to reboot into 40-columns to run the TextPro macro. If you're using the Epson Stylus 800 or Stylus 300, I suggest loading the labels in the regular tray and not feeding them singly through the back. The front feeder handles the labels just fine and with less chance of each sheet being crooked or misfed.

Macro execution in TextPro isn't lightning fast. It's like running a program in BASIC instead of machine language. My TFLABEL.MAX macro uses all three TextPro banks to process the text. The macro could be optimized

to use fewer banks, and to make it faster. I was in a hurry and didn't take time to optimize the macro or to write a program written in a traditional programming language. It speaks well for TextPro's extensive macro capability that I was able to accomplish my task without writing a whole separate program.

Printer Control Codes

There are a number of printer control codes that TFLABEL.MAX puts into the address label document just before it prints it. A few TextPro PrintKey definitions are also made by the macro, but most of the printkeys have been predefined and saved as part of the TEXTPRO.CNF file, which is why you need it. For reference, the list below shows you what printkey definitions have been saved as part of TEXTPRO.CNF:

A=27	N=113	1=1
B=33	O=107	2=2
C=10	P=40	3=3
D=68	Q=52	4=4
E=69	R=53	5=5
F=130	S=83	6=6
G=70	T=84	7=7
H=229	U=21	8=8
I=64	V=77	9=9
J=25	W=80	0=0
K=11	X=120	
L=45	Y=103	
M=____	Z=112	

It would have been convenient for me to explain line by line the control codes TFLABEL.MAX sends to the printer to ease the task of modifying it. Unfortunately I don't have the space to do so. Use the above table as a reference when examining the macro to determine what codes are being sent to the printer. Some remarks in the macro itself will give you a general idea of what I did. TFLABEL.MAX uses all three TextPro banks to manipulate the address data. Thus, it uses two 130XE expanded memory banks. TFLABEL can't handle address files larger than about 14K. It was designed for small personal mailing label lists, not something humongous like the *AC* subscriber database.

Labels For Non-Turbofile Users

I wrote two extra macros for non-Turbofile users that work much like TFLABEL for Turbofile. These are NEWLIST.MAX and ULABEL.MAX. NEWLIST helps you construct your own address list while ULABEL works like TFLABEL to manipulate and print the data.

To use NEWLIST make sure that, as above, you have your subdirectories and search path configured properly. Then type "-NEWLIST" and hit RETURN. If you've already used the NEWLIST macro file and want to edit or add to an existing list of addresses, you can use "-NEWLIST filename[.ext]" instead. (If you do this, and the file isn't found, NEWLIST won't load properly. You'll have to press the [START] key and type "NEWLIST" to load and activate the macro manually.)

Once NEWLIST is loaded and running, you'll see an options menu along the top of your screen as follows:
- L - Load file.
- A - Add an address at the current cursor position.
- P - Save file, and then print, via the ULABEL.MAX macro.
- E - Stop execution of the macro, and drop down to the standard Textpro editor. If you do this, use [OPTION]-M to reactivate the NEWLIST menu.

- S - Save file

If you hit the space bar you'll see an additional menu of options:
- CTRL-UP-ARROW - Move up one address
- CTRL-DOWN-ARROW - Move down one address
- H - Home position, or top of document
- B - Bottom of document
- M - Redisplay the main menu

Pressing any valid key will do what the description says. Note that key commands from *both* menus are always active, even though you can only see one menu at a time.

NEWLIST is simple but really useful. I use it to print disk labels for my 3.5" disks on mailing label stock. Adding the printer control codes for EXPANDED print on the first line of each label makes each disk easier to locate. NEWLIST has no problem accepting these codes as part of an "address" entry. It's up to you to make sure your text fits on the label.

Non-SpartaDOS-X Users

For people who don't use SpartaDOS-X, using TextPro and its macros isn't so elegant. You can't pass parameters from the DOS command line prompt through a batch file to TextPro, and there's no subdirectory search path capability in other DOS systems.

With MyDOS 4.51 you almost get there, but not quite. MyDOS does have subdirectory support similar (from the user's point of view) to SpartaDOS-X, and its "Pick Directory" command does seem similar to the SpartaDOS-X search path. It isn't, though, and doesn't work nearly as well. You can, for example, put ALL your TextPro stuff in a subdirectory called "TPRO" on Drive number 4. Then use "R" from the MyDOS menu to set "D:" to be synonymous with "D4:TPRO:". After that, you can load "TPX-.COM" without any drive or subdirectory specifier. You'd think that TextPro would be able to find TEXTPRO.CNF, TEXTPRO.FNT, and TEXTPRO.MAX, which are the standard TextPro support files. It doesn't. TextPro will look on D1: for these files and give up if they aren't there. Go figure.

Once you've loaded Textpro as just described, manually load TEXTPRO.CNF with [SELECT][CTRL]-L and TEXTPRO.MAX with [CTRL]-V. Now you're ready for business. Press the [START] key, type "NEWLIST" at the prompt, and TextPro seems able to look to "D:" default drive and directory for the appropriate macro file. The NEWLIST macro will successfully find ULABEL.MAX when it needs to use it.

If you use Turbofile and TFLABEL.MAX instead of just NEWLIST.MAX, the macro the operations are similar to using NEWLIST.MAX. Use the MyDOS "R" command to select the default drive and subdirectory. Load TextPro, then manually load TEXTPRO.CNF and TEXTPRO.MAX. Then use [CTRL]-L to load in the Turbofile output file. Once the file is loaded, press the [START] key and type "TFLABEL" at the prompt. Things will proceed normally after that.

Disk-based versions of SpartaDOS (2.3e and 3.2d) are almost as bad as MyDOS. All TextPro files need to be on D1: whereas MyDOS allows you to use a second floppy drive or RAMdisk assigned to something other than D1:. Put all your TextPro files, macros, etc. into a subdirectory on D1:. Use the "CWD" command to change the working directory to the "TPRO" subdirectory. TextPro will suc-

cessfully find all the files and work okay.

With all other DOS's, all your TextPro files need to be on D1:.

Auto-Dating Your Letters

If you use TextPro 5.0X, SpartaDOS-X, and the R-TIME8, check out LETR2.MAX on *AC*'s February '94 disk. It's an automated letter-writing macro. Begin by placing ECHO.COM in one of your search path subdirectories. Make sure DATE.BAT from the *AC* disk is also in a search path subdirectory. Type "-DATE" and press [RETURN]. This will create a small text file containing the current time and date.

You need two other text files, called RETURNAD.TXT and NAME.TXT. RETURNAD.TXT should contain your return address, and NAME.TXT should contain a closing followed by your name. LETR2.MAX works much like LETR.MAX, which came with an earlier full distribution of Textpro. LETR2.MAX will load the time/date file into one of the extra banks, erase the unwanted material, and incorporate the date as part of your letter. It's a great example of the kind of automation Textpro 5.0X and SpartaDOS-X can offer.

PrintShop and the XEP80

Daniel Carrodano of Roquebrune, France wrote me a very nice letter wondering if it were possible to print PrintShop stuff to an Epson-type printer using the XEP80 printer port. As I wrote in my December '92 column, the XEP80 doesn't have a standard printer port. The computer won't recognize the XEP80 parallel port without a special "handler" program being loaded first. There currently exists no such handler that works with PrintShop because PrintShop is a self-booting disk that doesn't load DOS and doesn't allow loading of a standard AUTORUN.SYS type program. I'm highly pessimistic about a PrintShop XEP80 patch ever being written.

However, Print-Filer from DGS in England should work. This program lets you send your PrintShop printouts to a disk file. Once there, you should be able to boot a DOS diskette with the Atari XEP80 AUTORUN.SYS handler present, and use the DOS "C" copy function to copy the PrintShop output file directly to the "P:" device. Print-Filer costs $15.00 including shipping and handling.

Sources:
Dean Garaghty Software
62 Thomson Ave.
Balby, Doncaster
DN4 0NU ENGLAND
+44-302-855-026

Epson Stylus 800 refill kit
$8.90 for ink, $1.25 for Syringe, $2.35 S&H
V-Tech Inc.
Hatfield, PA 19440 USA
215-822-2989

Exploring The Wild FONTier
by David Richardson
AC Staff Columnist

Up until now I've talked about how to run DD3. I've talked about the software commands, what settings to use, and how to get around certain problems. I haven't really mentioned much about how the hardware figures in all of this, and what sort of hardware might be the best to use. The best place to start is the minimum DD3 requirements.

Any Old Thing Will Do

DD3 will work with any Atari 48K 8-bit computer. This includes the 400, 800, 600XL, 800XL, 1200XL & 130XE. It will work with Atari DOS 2.5, MyDOS, SmartDOS, SpartaDOS 3.2 or SpartaX. According to the DD3 documentation, DD3 will work with the following printers or their compatibles:

- Epson EX/FX/JX/LX/MX/RX printers and compatibles.
- BlueChip 120/10, Legend 1080, Mannesmann Tally Spirit 80
- Star Gemini 10X/SG10
- NEC Prowriter/C.Itoh 8510A
- Atari XMM801

If your printer can emulate any of these printers, then you can use DD3. My dot matrix printer is a Star NX1000, which is Epson compatible. It works great. If all you have is an old 800 and one 810 single density disk drive, DD3 will run on your system. Of course, different hardware configurations can make DD3 a lot easier to use, and a lot quicker. If you have only one single density drive, then you can only use text files and fonts that will all fit on the same disk. Since some fonts, such as the graphic fonts, take up a lot more room than others, you will not be able to have as many fonts in one printing as you will if the fonts are smaller. A double density drive will allow you more space to put either larger fonts or more of them. If you have two disk drives, you can put some files on one, and some on the other, but you must make sure that the text file you are trying to print has the appropriate drive designation to go with the font change command when it uses a font on the second drive. Of course, if you have a RAMdisk, this gives you more drive space, and will speed up the process considerably. Obviously, the larger the RAMdisk, the better.

When I first started using DD3, I had an MIO from ICD that had one megabyte of memory in it. The MIO had no battery backup, so I left it on at all times. I even had it set up as the number 1 drive, so I was booting up from it. It was pretty nice to have all of that storage space and speed. It pretty much spoiled me. I hadn't realized how much disk access was needed for DD3 to run. One day I wanted to see how long it would take to print a page out of my user's guide using floppys only. I chose a page that had a lot of font changes and was almost completely covered with graphics. It took 45 minutes to print that one page. I had been used to it taking only about 5 minutes or so for that particular page. This, of course, was a worst case scenario. Most people will use DD3 to print text, and usually they stick with the same font. It was clear to me, however, that faster disk access was the best way to go. Since I was accumulating more and more DD3 fonts, I was out-growing my MIO. That was when I decided to use a hard drive.

What's In That Black Box?

For those of you who may not know, the Black Box is an interface that you hook up to an XL or XE computer. It will NOT work with an 800 or 400. It will allow you to hook up a printer, modem, and hard drive to your Atari. The Black Box is available from Computer Software Services. They advertise in Atari Classics. In the December 1993 issue of AC, the CSS advertisement has the Black Box for $179.95, not including the hard drive. A good rule of thumb is that you can usually figure to pay one dollar, perhaps a little more, per megabyte for the hard drive. Just make sure that you talk to CSS to get a drive that is compatible. To use the Black Box, you will need to use either MyDOS or Sparta. For the Atari, you won't need a real large hard drive. Mine is 80 meg and it's plenty large enough.

Another thing about the Black Box is that for $150 more, you can get a Floppy Board added to the Black Box that will allow you to use IBM floppy drives with it. You can get a 3.5 inch high density floppy drive for about 60 dollars. I tend to use these floppys rather than the hard drive anyway. Personally, I think that this is probably the best way to go. For one thing, if your hard drive was to ever crash, and you were really depending on it, you would really be hurting, especially if it is damaged and you need a new one. It's pretty easy to forget to backup some data, and then lose it because you have hard drive problems. If you ever need to replace the floppy drive, you only need another 60 bucks. Also, since the Atari doesn't really need THAT much space, its pretty nice to simply use different floppys for different uses. I've got a 3.5 floppy I boot for DD3 and TextPro, and I put everything else on other disks. Besides, the 3.5 inch floppys are real fast anyway, much faster than Atari floppys. If you use the 3.5 floppys first, you will probably feel no need to go to a hard drive. And, with your programs scattered among lots of floppys, you won't lose everything with one swoop like you would on a hard drive you forgot to back up. Also, if someone you know also has IBM floppys, and you want to go to their house to do some computer work, you simply pack up your floppys and go.

No matter what hardware you use, remember that DD3 is constantly reading from whatever drive you are using as it prints. Keeping this in mind, you can see that the best way to go is the one that is fastest, and has the most amount of storage space. I can't think of any example on the Atari when you'll need drive space larger than on the IBM floppys, except perhaps graphics such as GIF files. Also, if you are like a lot of 8-bitters, you are probably concerned about repairs on your 8-bit hardware, especially the disk drives. Extensive use of DD3 might hasten the demise of your 8-bit floppy drive. The problem isn't really the cost, but the availability, of 8-bit replacement drives or parts. Where will you get them, if at all? If you use an IBM floppy and it shoots craps, all you do is go to your local computer store and buy another. I go for months at a time without even turning my 8-bit drives on. With this setup, you can save your 8-bit floppy drives for programs that HAVE to be run from them, such as PrintShop, or the huge group of games that are dedicated to a bootable disk, and not file-oriented, such as Lode Runner, or any other commercial software done

in this manner.

Another reason I tend to favor using IBM floppys as opposed to the hard drive is, let's say you get the Black Box with no floppy board, and a hard drive. Whenever you put anything on the hard drive, you must use the Atari floppys to start with. Suppose you fill the hard drive and decide to back up files so you can delete them from the hard drive. Perhaps you even delete certain programs. Any time you add files or remove files from the hard drive, you have to use your Atari floppys. If you use it for a year, that's a year of using the Atari floppys. Eventually, you'll realize that even though you have the Black Box and a hard drive, you are still using the Atari floppys. You may have deleted a bunch of games to make room for other stuff. Then later on, you want the games back on the hard drive. Pretty soon, you realize that you're still using the floppys to put files on the hard drive that you put on several times before. After a while, you'll still be wearing out your Atari drive. If you had gotten the floppy board instead of the hard drive, all of the files that you have been deleting the past year would have been backed up to IBM floppys, which you simply set aside rather than delete them. Then when you want them back, you simply insert the appropriate IBM floppy in the IBM drive. You could have went the whole year without even turning on your Atari drives. Keep in mind that as time goes by, it'll be harder and harder to get parts or replacement drives. All the years you use the Atari floppys begin to add up. This becomes more apparent if for some reason you have to get another hard drive, and have to use floppys to fill the new hard drive. With the IBM floppys, this problem is avoided. The only time you would need to use the Atari floppys would be when you get a new program that you hadn't put on the IBM floppys before. How often do 8-bitters get new programs? Plus, if you get programs by using bulletin boards, you save them to the IBM floppys rather than the Atari floppys.

I Didn't Know You Could Do That!

One thing that you might not know is that you can hook up a laser printer to the Atari. I have an Epson laser printer called the Action Laser 1500. It has built-in Epson FX, LQ, etc. printer emulation. By using the control panel on the printer itself, you can set it up to think that it is an old Epson dot-matrix printer, and then you can hook it up through either its serial or parallel printer port. It will remember how it was set up even when it is turned off, so you won't have to re-set it every time you use it. And, since it has two ports, you can use one to hook it to an Atari, and the other to hook it to another computer, such as an IBM. I own both, and my laser printer sits between both of them. The Action Laser 1500 is Hewlett Packard III compatible, so with it, I can use virtually any software on either the Atari or the IBM and it will work. You can even program the printer so that one port emulates the Epson FX/LQ series, and one port emulates the HPIII. The column you are now reading was done with it. If you have been thinking about getting a new printer, you might want to consider getting one of these.

SIO2PC

Another thing you might try is something called an SIO2PC, which is a combination of hardware and software. The hardware is an interface that consists of a box with two cables coming from it. One the end of one of the cables is a connector that is the same as those on Atari IO cables which you use to connect your floppy drive to the computer. At the end of the other cable is a serial connector which you connect to an IBM serial port. The box between these two cables consists of the electronic components that make it work. The software you get is on an IBM disk which you use to program your IBM into accepting the role as a slave to the Atari. Basically, this turns the IBM into a peripheral for the Atari. You can then use anything hooked up to the IBM for use for the Atari as you would any normal Atari device. You can save files to IBM floppys or hard drives, you can send to the printer hooked to the IBM, and you can even use the memory in the IBM as a RAMdisk for the Atari. The IBM is then an extension of the Atari. It will work with any Atari computer, even the 400, assuming the 400 has 48K. It will work with any IBM compatible computer, even the old 8088 computers.

For more info about SIO2PC, read the following reviews in AC:

Feb. '93 by Mike Jewison p. 7
Feb. '93 by Alan Sharkis p. 13

Or, you can write to the creator of SIO2PC:

Nick Kennedy
300 South Vancouver Street
Russellville, AR 72801

Final Analysis

I think that the people who can most benefit from getting the right hardware are those who have an Atari computer only. By spending about $400.00 (the price of the Black Box, a Floppy Board and an IBM floppy drive), you can really extend the life of your Atari and you'll be able to save wear and tear on your Atari drives. So, for those Atari die-hards who absolutely refuse to give up their 8-bits, this is the best way to withstand the onslaught of the IBM invasion. Even if the time comes when your Atari computer finally bites the dust and you have no choice but to get an IBM, the 3.5 floppy drive and the laser printer can be used on the new system.

I hope that this column has helped everyone out there, if for no other reason than to make your life easier with the Atari 8-bit computer.

Keep on trekking.

David Richardson
P.O. Box 746
Lawrence, Kansas 66044
913-843-5213

Floppy Drive Mods And Repairs

W/P Stuff

Although no longer in production, the Atari 810 and 1050 drives are still probably the most common storage devices in use today by us 8-bitters. As a result, there have been several commercial and home-brew modifications for these drives. One that I had never seen, however, was the addition of a Write Protect Light. A W/P light indicates if a disk is write protected or not. My Rana drives have W/P lights. Although they are not a necessary item, I have really come to depend on mine to help avoid mishaps during operations that require heavy disk swapping. W/P lights can also be found on Indus and Trak drives.

This month's column deals with the installation of a W/P light and Write Protect Switch to your 1050 drive, as well as tips for common repairs to various Atari-compatible drives. The W/P switch lets you select any one of three different Write Protect modes at any time. (There's no need to turn power off or even wait for the "BUSY" light to go out when switching modes.) Be aware though, if the drive is switched to the PROTECT mode during a write operation, an error will be generated and the write operation will be aborted. The three modes are:

Mode 1: "UNPROTECT" This mode allows you to write to un-notched disks or disks that have Write Protect Tabs on them. It will also allow you to write to the "flip" side of disks without having to cut a newnotch.

Mode 2: "PROTECT" This mode will force the drive into Write Protect mode even for notched disks. All disk writes are disabled.

Mode 3: "STANDARD" This mode sets the drive for normal (unmodified) operation. Only notched disks can be written to.

Although either modification could be done separately, you can add them both for only about $5.00. Together they'll give you 100% control over the drive's write protect function along with a visible indication of the write protect status.

What's Required?

As with all hardware projects, you'll have to begin this one with a quick trek to your local electronic parts supplier. Since for many readers that means Radio Shack, I've included RS catalog numbers for your convenience. Purchase the following items:

1.) A SPDT 3-position "ON-OFF-ON" switch (#275-325);
2.) A 330-ohm 1/4 watt resistor;
3.) A light emitting diode or "LED" (#276-026);
4.) Approximately 24 inches (60cm) of insulated hook-up wire;
5.) Rosin-flux solder.

With parts in hand we're now ready for business. I'll as-

by Paul Alhart, AC Staff Columnist

sume you already have a soldering iron and the usual assortment of small hand tools.

Getting Started

Before you even remove your drive from your system, stop and look things over. Decide where the best location for the switch and LED will be. Since you'll have to drill mounting holes in your drive's case, you have to get this right the first time. On the 1050 drive, I found the best location to be on the downward sloped bevel on the right side of the front panel. Wherever you decide, make sure there is clearance inside the drive case for your components before drilling any holes.

Refer to the schematic diagram in Figure 1 below for installing the W/P switch and LED indicator in your 1050.

Going Inside

Remove the drive from your system and flip it over. Remove the six screws that hold the case together. Flip the drive back right side up and lift the top half of the case off. The front panel usually comes off with the top case half and can just be unsnapped from it if desired. Before going any further, get a paper and pencil and make note of the orientation of the seven connectors linking the drive mechanism to the printed circuit board (PC board). There are five connectors on the left rear side of the board, one on the right rear side and one on the right front side.

These connectors will have to be removed, and if you reconnect them backwards or in the wrong order you could cause permanent damage to your disk drive. Using a small pair of pliers, grasp the plastic portion of the connector (NOT THE WIRES!) and pull it straight up off the
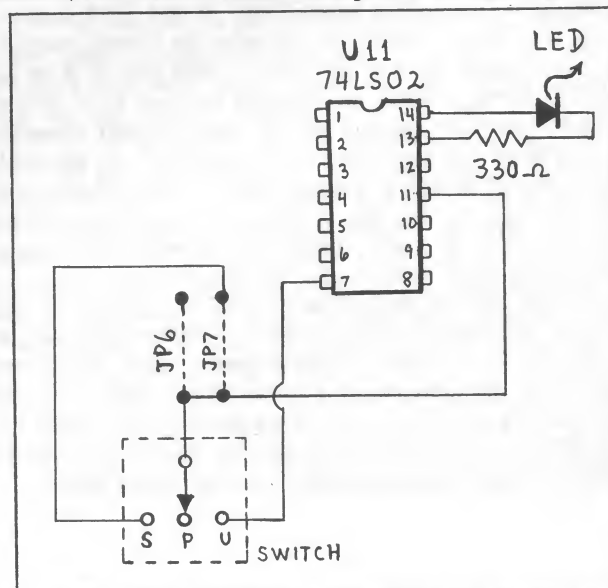
Figure 1. W/P switch and indicator modification for the Atari 1050 drive.

PC board. **CAUTION**: Most problems encountered when performing mods to 1050 disk drives occur at this point. The wires can easily be ripped from their connectors if they're yanked too hard. Easy does it! After all the connectors have been removed from the PC board, lift the drive mechanism straight up and out. Grasp the drive mechanism by its sides. Never lift or hold it anywhere near the head assembly. Set the drive mechanism aside in a safe clean place.

ESD Caution

Although you won't be removing any ICs, you'll be working directly on the PC board. Take normal precautions against ESD (electrostatic discharge). If you don't know what I'm talking about here, STOP! There have been many good articles written on the subject of ESD. Read one of them before you proceed. If necessary, I'll be glad to cover ESD further in a future article. Many of the components in your system are static-sensitive and could be damaged if you're careless in this area.

Spread the two plastic clips in the lower case half that hold the front of the PC board in the case. Remove the PC board from the case. Remove the RF shield from the PC board by straightening out the four metal tabs on the bottom side of the PC board. Locate U11 (a 74LS02 NOR gate) and JP7: they're on the PC board in the area under the RF shield you just removed. You should be able to easily spot the row of jumpers including JP7; U11 is the closest component to the jumper.

Installing The W/P Switch

If you're installing the W/P switch, remove JP7. Locate the solder pad of JP7 that's common with the pad of JP6. This is the pad toward the rear of the PC board. From this point, connect a wire to the center contact of your W/P switch. Connect a wire from the other JP7 pad to one of the W/P switch's end contacts. Connect a wire from the contact on the other end of the switch to U11 pin 7 (GROUND).

All that's left to do now is mount the switch. I found that positioning the switch so that UNprotect was on the left and Standard was on the right made its operation easy to remember. Just think of the shipping outfit, UPS, when making a selection. I guess "Uninterruptable Power Source" would do just as well!

Installing The W/P Light

Connect U11 pin 13 to a 330-ohm 1/4 watt resistor. Connect the other end of the resistor to the cathode of the LED. The cathode is the longer of the LED's two leads. Connect the other lead of the LED (the anode) to U11 pin 14 (+5V). Mount the LED and you are done. You might need a small dab of RTV silicone or glue to hold the LED in place.

Reinstall the RF shield. Note that the notched portion of the upper half is over U14 and R43. Insure that the added wires aren't pinched by the RF shield. Install the PC board in the lower half of the case. Set the drive mechanism back in place on its mounting pins. This can sometimes be the most challenging part of the whole project! Now plug the seven connectors back onto the PC board. Again, be careful here. Be sure they're positioned correctly. Replace the top case half and front panel, flip the drive over, and replace the six screws.

Testing

The drive is now ready to test. Before placing it back in service, connect the power supply and turn power on. The busy light should light for a few seconds during which time the head stepper motor and drive motor can be heard. If the drive motor doesn't run or won't stop, or the busy light doesn't light or won't go out, turn power off. Recheck all the connectors to ensure they're positioned correctly and that none of their wires have been pulled loose. After insuring everything is OK to this point, put the drive back on line. With the new W/P switch in the center position the W/P light should be ON, indicating PROTECTED mode. Flip the W/P switch to one of its end positions and insert a write protected disk. If the W/P light comes on, this is the STANDARD mode. If the light doesn't come on, it's in UNPROTECT mode.

Both of these modifications are the type that, after a few weeks, you'll wonder how you ever got along without them. If you feel uneasy about opening your drive and poking around with a soldering iron, find someone qualified to help you. As a service to Atari Classics readers I offer installation of both 1050 W/P upgrades for $25. This includes parts, installation, cleaning and alignment of the drive mechanism, and return shipping. To take advantage of this offer, or if you'd like to see similar upgrades published for other disk drives, let me know! (My address is at the end of this article.)

Drive Weirdness

Having repaired hundreds of disk drives for the 8-bit Atari, I thought I'd seen just about every type of failure and kludge there was. There were the usual failures due to dirt, misalignment, and faulty power supplies. And then there were the oddballs, like the 1050 that had a can of cherry cola dumped into it. Some of the user fixes I've seen were pretty ingenious too, like the Rana that had a guitar string used to replace the head positioning band. (A better solution would have been to use the band assembly from an Atari 810 MPI drive mechanism. Atari part #FA100034, from Best Electronics. This is a direct replacement for the band assembly used in the Rana.)

Belting Your 1050

Recently a new problem has been showing up in 1050 drives with the World Storage drive mechanism. The drive belt keeps falling off for no apparent reason. When this happens the disk stops turning. Even though the busy light comes on and you can hear the motor running, the drive won't read or write to any disk. If this is happening to your 1050, fear not. The explanation is fairly simple, and the fix is even easier.

In what I consider a design flaw, World Storage failed to include a belt retainer on their drive like Tandon did. This wasn't a problem when the drives were new. Now, however, a few years later, when the spindle bearings are starting to wear, it's becoming a problem for many 1050 owners. The slightest tilt of the spindle due to sloppy bearings causes the belt to track right off the edge of the pulley. Atari didn't switch from Tandon to World Storage drive mechanisms till near the end of 1050 production. This explains why the problem is just now starting to show up.

As I said, the fix is easy, and it won't cost you anything but a few minutes to make it. Just follow the procedure outlined below to add a belt retainer to your 1050.

Step 1: Disconnect the drive from your system and flip it over.

Step 2: Remove the six screws holding the case halves together. (Four are in recessed holes in the case.)

Step 3: Flip the drive back right side up and lift off the top half of the case.

Step 4: Lift the drive mechanism from the front without putting any undue stress on the wires running from the drive mechanism to the controller board.

Step 5: Locate the spindle pulley. If it's fully exposed to view you have a World Storage drive mechanism and should proceed with the fix even if you haven't had a belt problem yet.

Step 6: Cut a disk from the top of a plastic butter or Kool-Whip tub. Use a compass or drinking glass as a guide to make the disk about 1/4" larger in diameter than the spindle pulley.

Step 7: Glue this plastic disk to the bottom of the spindle pulley. Hot-melt glue works really well for this.

Step 8: Reinstall the drive mechanism by setting it down on its guide pins.

Step 9: Replace the top cover and the six screws removed in step 2.

Step 10: Reconnect the drive to your system. The plastic disk acts as a belt retainer and should allow you to enjoy many more productive years from your 1050.

Wake Up Your 810

Does your 810 occasionally turn itself off or refuse to power up? This is a fairly common problem that can usually be traced to a burnt or corroded pin on the Side Board connector. This is the 23-pin connector that may be labeled J101 or J106, depending on which version of the 810 you have. Follow the steps outlined below to "wake up" your sleepy 810.

Step 1: Disconnect the drive from your system.

Step 2: Remove the four screws holding the top case half. These screws were originally hidden under circular plastic covers held in place by rubber cement. If your drive still has the screw covers, pry them off with a knife blade to gain access to the screws.

Step 3: Lift off the top case half.

Step 4: Remove the five screws holding the drive mechanism in the lower case half.

Step 5: Remove drive mechanism from the lower case half by lifting it straight up.

Step 6: Remove the three screws that hold the Side Board in place.

Step 7: Make note of the orientation of all connectors attached to the Side Board and carefully remove each one. NOTE: pull only on the connectors. DO NOT pull on the wires.

Step 8: Remove the Side Board by gently rocking it end to end, while pulling upward.

Step 9: Examine the 23-pin connector for signs of corrosion or discoloration. Extreme cases may show signs of melting or distortion of the plastic. Also examine the pins this connector mates with. Pay close attention to pins 1, 2, 10, 12, 13, 14, and 15. These are the pins that carry the most current.

Step 10: Clean any corrosion with contact cleaner or alcohol. An old toothbrush will help here. A pencil eraser will work well for the mating pins. Extreme cases may require replacement of the connector, which I won't get into here.

Step 11: Reinstall the Side Board, making sure the pins are aligned properly. It is possible to be off a pin on either end, so double check before reinstalling the screws.

Step 12: Replace all the connectors in their original positions as noted in Step 7.

Step 13: Replace the drive mechanism back into the lower case half and replace the five screws removed in Step 4.

Step 14: Replace the top case half, its fours screws and screw covers.

Step 15: Reconnect the drive to your system and test.

Help Your Trak Quit Smoking

On some Trak drives the power supply connector can make intermittent contact with the metal back plate of the case. This causes the internal bridge rectifier to go up in smoke. To eliminate this problem, remove the case and make the access hole in the back plate bigger with a drill or file. Be careful not to let metal debris get into the drive mechanism or the electronics. I don't have a Trak drive available to look at as I write this, so you'll have to figure out how to remove the case on your own.

Indus Owners Beware!

In the beginning, and through the 1200XL, virtually every Atari product used a 9-volt AC power pack. The small 1/4" diameter connector on this power pack is easily distinguished from the larger 7-pin DIN connector used on the +5 volt DC power packs of the XL/XE line of computers. Crossing the 800's power pack with the one from your 810 causes no problems because both devices use the same 9 volt AC power. These power packs are also interchangeable with those used by Trak and Rana drives as well. NOTE: some of the earlier Atari power packs had smaller wattage ratings and may not be appropriate for heavily expanded or modified equipment.

The Indus GT power pack looks the same as the Atari power pack. It even has that same 1/4" connector that will plug right into your Atari, Trak, or Rana. This power pack, however, puts out 9 volts DC. Accidentally plugging an Atari AC power pack into your Indus will cause unwelcome clouds of smoke to fill the room. I recommend conspicuously marking your Indus drive and its power connector with a warning to this effect. This will help keep your Indus healthy, especially if the kids get to messing around with your system.

Rana Guide Available

I've written a useful booklet called *The Rana Repair Guide* for Rana owners. For more tips on repairing and modifying your Rana, as well as complete schematics and alignment instructions, order your copy today. It's just $10.00 +$2.00 P&H. Send your order to:

Paul V. Alhart
524 North Zee St.
Lompoc, CA 93436 USA

And let me know if you'd like to read more about the Atari disk drives!

MIDIMAX: MIDI FOR THE CLASSIC-8

REVIEW BY ALAN SHARKIS, AC STAFF REVIEWER
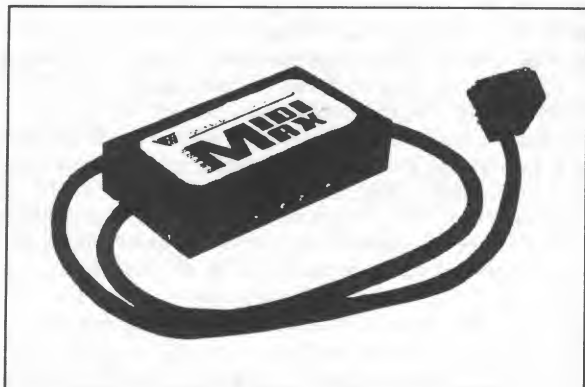
MIDI On The Falcon

The release of the Atari Falcon 030 has musicians cheering. After a usergroup meeting that includes a demo of the Falcon's MIDI capabilities to my club's musicians, I walk away with such phrases as "sixteen channels of digital sound directly recorded to hard disk" and "DSP port" ringing in my ears. Face it. I'm an 8-bitter who never experimented with MIDI while my computer was new, and my musical skills are, ummm... a little rusty to say the least. It was natural for me to try to catch up with all this Falcon excitement going on, but I wasn't about to buy a Falcon just yet. It even took me awhile to recall what MIDI meant: Musical Instrument Digital Interface. Some things come back to you slow-like.

At a recent meeting of LIAUG (the Long Island Atari User Group, one of several groups I belong to) a gentleman named Steve Cohen came down to represent his company, Wizztronics. Yes, the very same Wizztronics who made the 256K RAM upgrade reviewed by Mike Jewison in the April '93 *AC*. They now make innovative products for the ST and MS-DOS machines, but they started with the Atari 8-bit computers. He brought along one of the last complete packages of MidiMax, an 8-bit product his company had produced some years back. To make a long story short: I came, I saw, I bought.

MIDI For The Classic-8

The MidiMax kit contains an interface that plugs into the SIO port and gives you the typical MIDI IN and OUT ports. The interface is beautifully made. There's an additional SIO port on it so the unit isn't dead-ended. The case is steel with a rugged wrinkle finish and is riveted together.

Whenever AC-powered music equipment is connected, you always have a slight chance of a high voltage appearing where it shouldn't. A potential (pun intended) exists for one unit to fry another. So the makers of MIDI keyboards and other devices include optoisolators in their instruments to prevent catastrophes. The MidiMax interface contains the same optoislators used in more expensive keyboards. The kit also contains two MIDI cables and a simple sequencer program called Midi Music System (MMS) written by Lee Actor (of AMS fame) under the Synthetic Software label.

Hooking the interface up is easy. You plug the SIO cable into your daisy chain, then connect MIDI IN on the interface with MIDI OUT on your keyboard using one MIDI cable supplied. The second MIDI cable connects MIDI OUT on the interface with MIDI IN on the keyboard.

Quest For MIDIMAX Software

The MMS software worked just fine for what it was designed to do, but I found it frustrating. When I asked Steve some weeks later if his kits were still available, he told me the interfaces were still available but the software wasn't, unless he suddenly got a very large demand for it. I couldn't, in good conscience, review the product under those conditions. He did tell me that there might be some pieces of MIDI software for the 8-bit still floating around that had been released by Hybrid Arts under the name of MidiTrack I, II, and III. They'd work with the MidiMax interface, but they were different from MMS in several respects (I'll get to that later). How could I get MidiTrack? Steve didn't know at the time.

A few weeks passed. I attended the Connecticut AtariFest and ran across Steve Cohen again—and pressed him once more for any leads on MidiTrack software. This time I was in luck. Hybrid Arts had closed up shop and transferred rights and inventory to a company called Barefoot Software. Barefoot had a booth at the Connecticut AtariFest, right next to the LIAUG table. Unfortunately, they had none of the software, but had sold it all to B&C ComputerVisions. I had no idea of the time frame. I couldn't recall seeing that software listed in a B&C ad in recent months. My software quest seemed futile, so I went to work with MMS just to satisfy my curiosity.

MIDI Music System

MMS turned out to be a "step-time" program. You enter notes from your MIDI keyboard or computer keyboard, and they get stored in a buffer. You then press RETURN to get each note listed in a "track." The time value of each note (quarter note, half note, etc.) has to be changed manually each time the music calls for a change-tedious! The tracks, moreover, are monophonic: they can't include chords. It's a fine system for beginners and those who wish to do a minimum of work but want their keyboards to play from their 8-bits.

MMS also comes with a conversion program, written by Jim Thompsen, that converts AMS songs to MMS. That can keep many of us who aren't musicians busy for quite a while. After you read in the AMS file and the conversion program writes out the .MUS file, you load the .MUS file into MMS, choose some instruments, sprinkle in a few other MIDI commands according to the capabilities of your system, and your keyboard will serenade you for as long as you like. I demonstrated it for my other user group, the Ol' Hackers (OHAUG) and they were duly impressed. Alas, this software just isn't available any more, unless, as Steve Cohen said, there's a big demand.

Enter *Atari Classics*

Atari Classics provided the key that made writing this review possible. In the June '93 issue, right on there on page 15, was the B&C ComputerVisions ad: in that ad was a listing for Miditrack II and III, complete with inter-

---

faces. I had the interface already and didn't need another one. I called B&C and asked if it was possible to get just the Miditrack III software. The people at B&C had just received the shipment and had to open the carton to see if it contained any loose software. I was in luck: they had some pieces consisting of just the software and manual. I ordered one, and I've been having all kinds of fun with it ever since!

Introducing Miditrack III

Miditrack III was written especially for the 130XE computer. Miditrack II runs on unmodified 800's and 800XL's and is probably less elaborate. I know from the Miditrack III manual that Miditrack III will import Miditrack II songs, but not the other way around. The file formats are different, and Miditrack III allows storage of longer compositions than Miditrack II.

For some reason, Miditrack III won't work properly if I have a cassette drive in the daisy chain: the keyboard locks up. I don't think that will affect too many people, since cassette storage is all but dead in this country. I just disconnected the cassette and had no more problems. Possibly the conflict is the result of my having added

CSS's US+ operating system to my 130XE, but I can't be sure.

Another word of warning is to read the manual before you use this product. This program is loaded with features, but the keystrokes to get to them are complex. The manual does provide a command summary in Appendix B.

It's time to take some of you to school. You're all no doubt familiar with mechanical music. Is there anyone who hasn't had some experience with a Jack-In-The-Box toy? You know, you turn the crank, music plays, and at a predetermined time, the lid springs open and a clown head pops out. The music always plays in the same order (or the reverse order if you turn the crank backwards.) You can slow it down or speed it up, or stop in the middle, but the order and pitch of the notes and the time "Jack" pops up in relation to the music is always the same. The whole thing is digital: either on or off. A pin on a barrel hits a steel reed or it doesn't. It's a primitive example of a sequencer.

Under the MIDI standard, everything is also digital, only it's electronic. Clock pulses get sent 24 times each quarter-note to keep everything synchronized. Messages consisting of bytes tell the instruments in the hookup
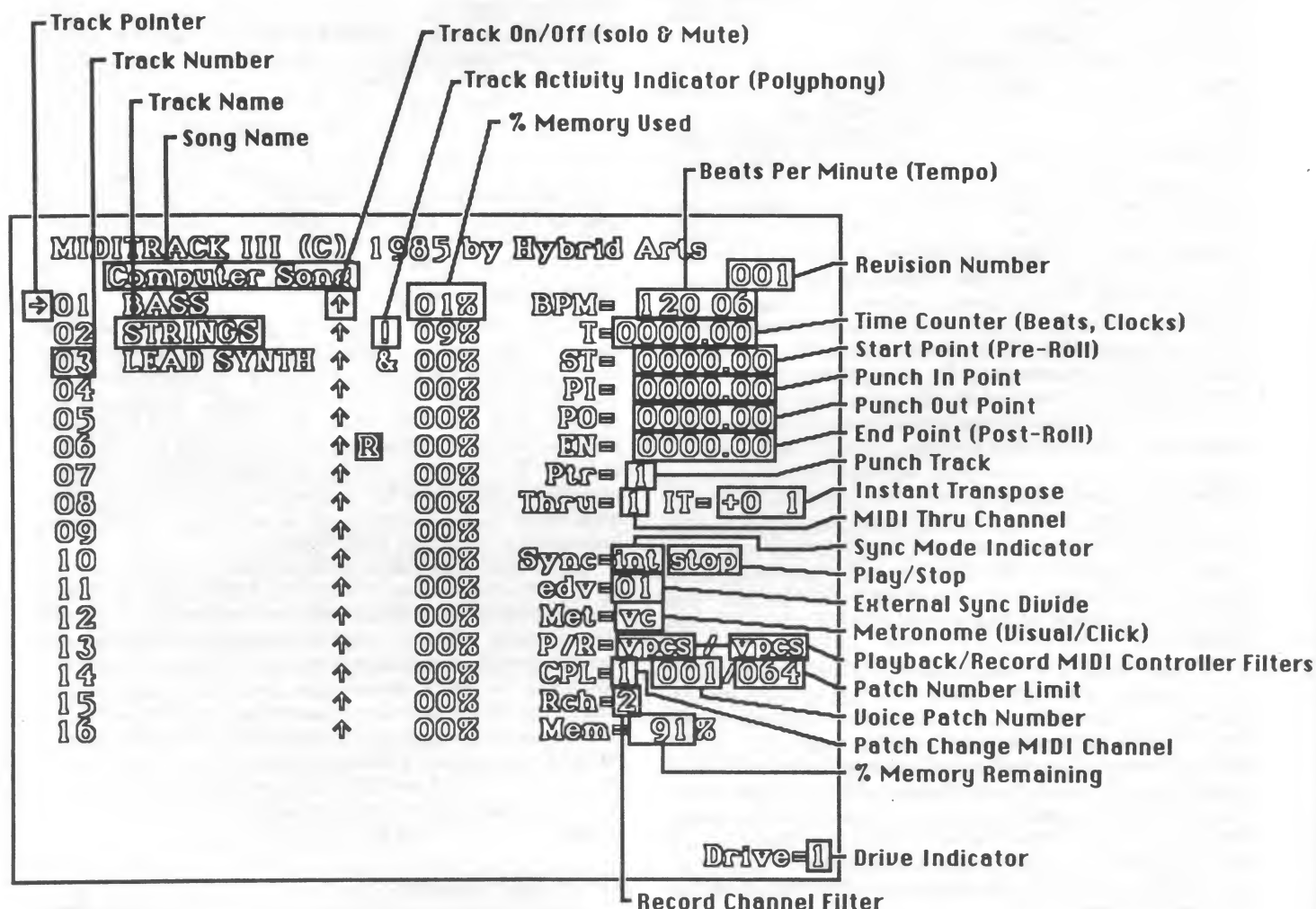
The MidiTrack Screen

what note to play or not play, how long, how loud, with which voice. The messages come down the line serially at 31.25 Kbaud. The sequencer, which can be built into a keyboard, stand-alone hardware, or a computer program, puts those messages into order and transmits (playback) or receives (record) them in order. The user can also manipulate those messages, or MIDI events, add to them, edit them, combine them, etc., depending on the complexity of the sequencer. MidiTrack III permits you to do these things with your Classic Atari and has many advantages over MMS for more accomplished musicians. I'm going to try to whiz through this, so I'll ask you non-MIDI folks to please bear with me as I give you a lightspeed tour of the software.

Using Miditrack

Recording is usually a real-time affair, but step-time is also available. Sixteen tracks are available, and any track can be assigned to any of sixteen MIDI channels. Each track can be individually titled, as can the song name. Tracks can be soloed or muted. Track-mute presets are available. Tracks can be write-protected in RAM. They can be deleted, moved, copied, jumped and looped. They can be quantized, combined, and uncombined. Overall velocity and relative velocity changes can be set for each track, as well as overall note duration. It's easy to locate and eliminate hung notes.

Tempos are adjustable from 2 to 750 BPM, and preset tempos can be stored in three registers. Step values are adjustable between 1 and 96. A time counter and start point for playback or cue are always on the screen (see the MidiTrack screen on page 30).

A song position pointer can be enabled. Start search makes it possible to send MIDI clock signals before start point so that sync can be locked. Sync, by the way, can be selected as internal, external, single step, MIDI or as TTL pulses through a joystick cable for non-MIDI drum machines. Punch-in and punch-out points can be set. Instant transposition is available, as well as a MIDI-Thru channel. A metronome is available with audible or visual clicks, or both. Playback and recording can include (or not include as the operator wishes) voice patch changes, pitch-bend, MIDI controllers, and Start/Stop/Continue commands. Aftertouch is recorded separately. Voice patch selection includes choice of channel on which the information will be sent, the voice number itself, and the upper patch number limit. Memory usage for each track and total is reported in percent.

Test tones can be sent over one or all MIDI channels. The full range of MIDI modes can be selected for each channel. For those MIDI devices that will respond to them, an All Notes Off command can be sent, a Notes Off can be sent for all notes currently sounding, pitch-bend and modulation wheels can be zeroed, and Local On and Local Off can be sent. Although notes aren't normally displayed on the screen, polyphony is, and the last note played can be displayed at the expense of audible playback on command.

A full range of disk operations is possible, and the program itself can be copied with a special command. You are encouraged to make back-up copies; up to four drives are supported. Unfortunately, an entire disk side is reserved for each song recorded. I've only skimmed the surface of MidiTrack's features; a single review really can't do justice to the power of this software.

Limitations Of 8-Bit MIDI

Probably the most serious disadvantage of MidiTrack III is its 40-column display. You can't view the main screen and the note display simultaneously, and when the note display screen is on, audible playback ceases. Also, many of the display items and the keystrokes to get them aren't intuitive. It takes a lot of time just to learn those keystrokes. Are the results worth the effort? I think so.

There's another whole set of problems introduced by the music keyboard you use. Mine doesn't respond to half the commands the program contains. I could remedy that situation by buying a better keyboard ($$$!). My keyboard has two banks of voices with 110 voices in each bank. I can't select banks from the Miditrack III software: that requires a system-exclusive message that Miditrack III won't send. Then again, neither will MMS. They were both written before there was much use of such messages. I have a simple composition program for my IBM clone that does send that message, but won't do half the things that MMS or MidiTrack III will do. So, either my compositions are limited to one bank, or I switch banks manually from my keyboard.

Both MMS and MidiTrack III also suffer from another disadvantage. They were both written before MIDI files were standardized, so their files can only be used on an Atari 8-bit running either of those programs. In fact, MidiTrack III song files, because they're so long, require another program to convert them for easy transfer on telecom systems. I believe that program might exist in the GEnie or CIS libraries but haven't looked into it.

The Bottom Line

The MidiTrack II (800/800XL/65XE) and MidiTrack III (130XE or RAM-upgraded XL) programs are available for $79.95 with interfaces from B&C Computervisions, 2370 Scott Boulevard, Santa Clara CA, USA; phone 408-986-9960 (see the B&C ad elsewhere in this issue). The Midi Music System software is no longer available, but you can order the MidiMax interface separately from Wizztronics at 31 Hewes Street, Port Jefferson NY 11777, USA; phone 516-473-2507. Using these products, 8-bitters with a musical inclination can take advantage of the capabilities available to them through MIDI. You might want to try this first before you run out and buy a Falcon!

MORE ABOUT COLRVIEW 2.6

BY JEFF POTTER, AC GRAPHICS & ENTERTAINMENT EDITOR

COLRVIEW Revisited

My thanks to all those who responded to the first article on COLRVIEW in the June '93 issue of *AC*. In this article I'll discuss where to find some COLRVIEW images, as well as what some other programmers (and myself) are up to with respect to new uses for COLRVIEW. I'll finish up by giving some technical details on how to write your own programs to experiment with COLRVIEW.

Finding COLRVIEW Pictures

The GEnie Atari 8-bit bulletin board recently moved all the COLRVIEW images and utilities to area 20, thanks to the work of sysop Craig S. Thom. Over 100 picture files have been uploaded there, courtesy of BB members. Over 50 of these have been uploaded by ANIMATSURI, also known as Wendell Hong. These are primarily Japanese Anime' images which Wendell has converted from GIF format, and are excellent at showing the clarity and saturation of colors offered by your 8-bit machine. My thanks to Wendell and the other contributors to this collection!

Also check out the Atari Star BBS at 305-868-0211 (7pm to 6am Eastern time) in the Miami, Florida area. Sysop Guy Ferrante has a fair collection of COLRVIEW pictures (in .ARC format) in file area 9, which he either downloaded or converted from GIF pictures. While you're at it, check out your local bulletin board system; if it's been around a few years it's bound to have a collection of COLRVIEW pictures.

To those Atari 8-bit owners out there who don't telecommunicate: I would seriously consider obtaining a modem. You'll discover a whole new world of applications for your 8-bit, and communicate with a multitude of other Atari owners. I wouldn't consider *my* system complete if *I* didn't have a modem! The direct-connect SX-212 is available very inexpensively (but you may have to hunt around to find one). Lacking that modem, you must obtain an Atari 850 or ICD P:R:Connection (or some other third-party serial bus to RS232 interface) and any manufacturer's modem. Excellent Shareware terminal programs exist (check out BobTerm 1.22). Don't put it off...

Other Software

E. Halliwell saw my COLRVIEW program some time ago and set out to correct some shortcomings in it. Namely, the three large files take up a lot of disk space, and there's no way for the viewer program to tell what resolution (GR.9 or GR.15) the image was stored in. So he created CVSQUASH, which reads in the .R/.G/.B files, lets you set the graphics mode, lets you add a caption, and stores one (smaller) file using a run-length encoding form of compression. He then created a companion "slide show" program (CSVIEW), with a friendly user interface, that lets you view these compressed (.RGB) files one after another with a programmable delay. Both programs were written in a mixture of BASIC and machine language USR calls, so the compressing and decompressing both run very fast. These programs are available in the GEnie Atari 8-bit files area (including source code), and are shareware. The author can be reached at:

Clay Halliwell
407 S. 2nd Street
Clinton, MO 64735-2107
GEnie: E.HALLIWELL

Adam Conover has created "Fractasia", a program that can create images of the famous Mandelbrot set. These images can be in any of several modes including COLRVIEW (you must use the original COLRVIEW program to view them, however). For those of you unfamiliar with fractal geometry, this program calculates an interesting equation in two dimensions which is like a world within itself with its incredible detail. Using this program, you can select an area of the screen to "zoom in", and recalculate. Each time you zoom in, you discover interesting nooks and crannies, and fascinating curling structures that are all part of one simple equation. No matter how fine you zoom in on the screen, more detail awaits you when it is plotted. Fractasia is written in compiled TurboBASIC, for XL/XE machines (must have 64k), and is available from the Internet ar-chives on atari.archive.umich.edu (see atari/8bit/frac1.arc and atari/8bit/frac2.arc), or directly from Adam. Fractasia is released as shareware. The author can be reached at:

Adam Conover
90 Plum St.
Greenville, PA 16125
Internet: A644690C@edinboro.edu
Compuserve: 70353,2030

The original APACVIEW was too large a program to attempt both GIF decoding and displaying in COLRVIEW mode. This was due to several (little-used) subroutines that tended to take a lot of room, and the general lack of memory on the old 48K Ataris. In my next program now in development, JVIEW XL, I will utilize the extra memory in the 64K and larger XL/XE series Ataris. This program will be able to read and decode GIF files and display them directly in COLRVIEW mode. You'll also have the option of saving the files in regular (.R/.G/.B) format, or compressed (.RGB) using routines borrowed from Clay Halliwell's CVSQUASH. And in response to several peoples' requests, JVIEW will also decode to GR.8 COLRVIEW, which has 320Hx192V pixels with eight colors per pixel.

Technical Details Of COLRVIEW

I won't delve into detail on how Atari display lists and display list interrupts work here. I'll refer you to one of several texts that include good descriptions (among them *De Re Atari*, the *Atari Technical Reference Notes, COMPUTE!*s series of books, etc.).

There are two ways to arrange the display list(s) to produce the COLRVIEW effect. One way is to store the color data in three RAM areas, separated by color. With this method you would need three display lists, and each display list would do a Load Memory Scan (LMS) instruction for every line. The LMS instructions within each display list would rotate among the red, green, and blue screen RAM areas. A Vertical Blank Interrupt (VBI) selects one of the three display lists at the start of each field. A display list interrupt (DLI) loads the color registers correctly to "paint" the correct color on a per-line basis. This is the way Clay Halliwell's CSVIEW creates the COLRVIEW displays; his source code is available on GEnie.

I considered using this method in my programs, but realized it used a lot of memory dedicated to display lists alone (over 1700 bytes). That, plus each display list must not cross a 1024 byte boundary in RAM (a limitation of the ANTIC display IC). Instead, I selected the other method: arranging the screen areas not by color, but in the order shown in the diagram from the first article (where the first screen goes RGBRGB, the second one goes GBRGBR, the third BRGBRG). Then I only need one display list which I modify in the Vertical Blank Interrupt (VBI) to change just the starting address and the midpoint address. I initially load the entire red, green, and blue data into three separate screens (7680 bytes each). I then "shuffle" the data to put it in red/green/blue "line" order. In this way, the display list can be quite simple, as I don't have to do a Load Memory Scan (LMS) on every line.

With either method, a Display List Interrupt (DLI) must be running constantly to change the background color (for GR.9) or all four colors (for GR.15) on every raster line. Although my method is a little harder to understand, I describe it here because it's the method I chose to implement. I'll include the subroutine I use to shuffle from one representation (red, green, blue screens in contiguous memory) to the other (first, second, and third screens in contiguous memory).

Bear in mind my discussion here is meant for the assembly language programmer. The subroutines presented here can be dropped in place around another program, but they can't stand alone. So, let's get started... .

Display List & Interrupt Routines

My display list has 16 blank lines at the beginning, the last one containing a DLI. Only one DLI is actually called for the entire screen, since it runs continuously during the display period (leaving no time for any other process to execute). This locks out things like

serial I/O, preventing you from doing neat things like loading another picture while you display one, or doing modem communication. Oh well, compromises have to be made in the name of 4096 colors!

I then do a single LMS in the selected graphics mode (9, 15, or maybe 8) and give the starting address of one of my three screens. The next 95 bytes can be ordinary display instructions. The display list breaks before the 97th line to do another LMS instruction. That's because ANTIC can't allow screen memory to cross a 4-kilobyte memory boundary. So I always arrange my screen memory to split nicely in half, with 96 lines before a 4K boundary, and 96 lines after one. After this we can include another 95 bytes of display instructions. At the end we do a "Jump on Vertical Blank" type instruction, and point to the beginning of the display list. Simple enough...

For the purpose of keeping track of which screen is being displayed, I created a variable called FLAG. This variable is incremented in the VBI, is clamped so it always remains between 0 and 2, and used as an index into a small table of screen memory starting addresses. I load the starting and midpoint addresses from this table, and poke them into my display list just after the two LMS instructions (at line 0 and 96). The VBI routine ends by calling the (immediate mode) system vertical blank routine (SYSVBV).

The DLI routines also depend on FLAG to tell them what the starting line's color is going to be. From that point on the DLIs rotate changing the appropriate color registers from red to green to blue in order. A line counter ensures that only 192 lines will be processed in this way, and the STA WSYNC ensures that the color register(s) are only changed during horizontal retrace, where you can't see it happening.

Loading And Shuffling

I'll assume you want to write a program to load or create COLRVIEW screens. The constants TOPSCX, TOPSCY, and TOPSCZ describe the starting address of the three color screens. You would load your three files into these areas (red, green, blue, respectively), call SHUFFL to shuffle them into the correct order, then turn ColrView on. After you're done viewing this picture, you would turn COLRVIEW off to disable all the VBI and DLIs. To restore the three color screens to contiguous blocks of red, green, and blue, merely call SHUFFL two more times.

Turning COLRVIEW On/Off

If you always load the same mode (GR.9, 15, or 8) files, you don't have to modify the display list. Otherwise you might have to write a routine to detect the graphics mode has changed and go through the display list to change the LMS and display instructions. See the listing of the display list for details.

Next you would JSR CVON. This routine first saves the current addresses of the display list and VBI routines, changes the ANTIC pointers to point to the custom display list, and sets the GTIA mode if necessary. It calls the system routine SETVBV, which ensures the vertical blank interrupt is correctly set in time for the next vertical blank interval. CVON then sets up the display list interrupt, and enables both VBIs and DLIs. Turning off COLRVIEW is done by calling CVOFF. This first turns off the VBIs and DLIs, so you can safely redirect the VBI pointers without fear of crashing. Then it restores the old display list and VBI routines, and turns off the GTIA mode. It then allows VBIs once again and exits.

Wrapping Up

These subroutines have been plucked right from my working source code, so they should work correctly. But in the course of transcribing and submitting for publishing, there's always a chance something won't work. If you get stuck trying to use them, drop me a letter or some e-mail. I'll do what I can to help.

I hope this helped some of you to gain a more thorough understanding of COLRVIEW. I hope to see more of you writing applications with it in the future. Good luck!

```
;********************************
; COLRVIEW subroutines for
; Atari Classics
; by Jeff Potter Aug. 17, 1993
; CIS: 74030,2020  GEnie: JDPOTTER
; Internet: potter@sunny.dab.ge.com
;
;********************************
; Display list, GR.9 COLRVIEW
; Caution: don't allow this to
; cross a 1K boundary.
; Same display list works for GR.8.
; For GR.15, change $4F to $4E and
; $0F to $0E below.
;********************************
;
DLLIST: DB $70,$30,$B0          ;16 blanks, DLL
        DB $4F                  ;LMS GR.8/9
DLTOP:  DW TOPSCX               ;starting adrs
        DB $0F,$0F,$0F,$0F,$0F  ;GR. 0/9
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
;
        DB $4F                  ;LMS GR.8/9
DLMID:  DW MIDSCX               ;mid-scrn adrs
        DB $0F,$0F,$0F,$0F,$0F  ;GR. 0/9
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
        DB $0F,$0F,$0F,$0F,$0F
;
        DB $41                  ;Jump on VBI
        DW DLLIST               ;back to top
;
;********************************
; Vertical Blank Service Routine
;********************************
;
MYVBI:  PHA
        TXA
        PHA
;
; increment flag byte
;
        INC FLAG
        LDA FLAG
        CMP #3
        BCC OK
;
        LDA #0
        STA FLAG
;
```

```
; Redirect display list to point          RED02   LDA  #$36              OUT8    LDA  #$00
; to appropriate screen area                      STA  WSYNC                     STA  WSYNC
;                                                  STA  COLPF0                    STA  COLPF2
OK      ASL  A                                     LDA  #$3A              ;
        TAX                                        STA  COLPF1                    PLA
        LDA  TOPSCR,X                              LDA  #$3E                      TAX
        STA  DLTOP                                 STA  COLPF2                    PLA
        LDA  TOPSCR+1,X                   ;                                       RTI
        STA  DLTOP+1                               DEX                    ;
;                                                  BEQ  OUT15            ;*******************************
        LDA  MIDSCR,X                     ;                              ; SHUFFL subroutine, called to
        STA  DLMID                        GRN02   LDA  #$D6              ; Swap color bytes around
        LDA  MIDSCR+1,X                            STA  WSYNC            ;*******************************
        STA  DLMID+1                               STA  COLPF0           ;
;                                                  LDA  #$DA             SHUFFL: LDA  #LOW[TOPSCX]
        PLA                                        STA  COLPF1                   STA  REDPNT
        TAX                                        LDA  #$DE                     LDA  #HIGH[TOPSCX]
        PLA                                        STA  COLPF2                   STA  REDPNT+1
        JMP  SYSVBV;exit(im.mode)         ;                              ;
;                                                  DEX                           LDA  #LOW[TOPSCY]
;*******************************                   BEQ  OUT15                     STA  GRNPNT
; Display List Interrupt Routine         ;                                       LDA  #HIGH[TOPSCY]
; Graphics 9, colrview            BLU02   LDA  #$86                              STA  GRNPNT+1
;*******************************                   STA  WSYNC            ;
;                                                  STA  COLPF0                    LDA  #LOW[TOPSCZ]
DLI1:   PHA                                        LDA  #$8A                      STA  BLUPNT
        TXA                                        STA  COLPF1                    LDA  #HIGH[TOPSCZ]
        PHA                                        LDA  #$8E                      STA  BLUPNT+1
;                                                  STA  COLPF2           ;
        LDX  #192                         ;                                       LDA  #0
        SEC                                        DEX                            STA  LINCNT
        LDA  FLAG                                  BNE  RED02            ;
        BEQ  THRED             ;0         ;                              TOPP:   LDY  #40
        SBC  #1                           OUT15   LDA  #0               ;
        BEQ  THGRN             ;1                  STA  WSYNC            ROW1:   LDA  (REDPNT),Y
        BNE  THBLU             ;2                  STA  COLBK                     STA  HOLD
;                                                  STA  COLPF2  ;black bgnd        LDA  (GRNPNT),Y
THRED   LDA  #$30                                  LDA  #$0A                      STA  (REDPNT),Y
        STA  WSYNC                                 STA  COLPF1  ;white chars       LDA  (BLUPNT),Y
        STA  COLBK                        ;                                       STA  (GRNPNT),Y
        DEX                                        PLA                            LDA  HOLD
        BEQ  OUT9                                  TAX                            STA  (BLUPNT),Y
;                                                  PLA                            INY
THGRN   LDA  #$D0                                  RTI                            CPY  #80
        STA  WSYNC                        ;                                       BNE  ROW1
        STA  COLBK                        ;*******************************
        DEX                              ; Display List Interrupt Routine  ;
        BEQ  OUT9                        ; Graphics 8, colrview          ROW2:   LDA  (BLUPNT),Y
;                                        ;*******************************        STA  HOLD
THBLU   LDA  #$80                        ;                                       LDA  (GRNPNT),Y
        STA  WSYNC                       DLI3:   PHA                             STA  (BLUPNT),Y
        STA  COLBK                               TXA                             LDA  (REDPNT),Y
        DEX                                       PHA                            STA  (GRNPNT),Y
        BNE  THRED                       ;                                       LDA  HOLD
;                                                 LDX  #192                      STA  (REDPNT),Y
OUT9    LDA  #$00                                 SEC                            INY
        STA  WSYNC                                LDA  FLAG                      CPY  #120
        STA  COLBK                                BEQ  RED8      ;0              BNE  ROW2
;                                                 SBC  #1                ;
        PLA                                       BEQ  GRN8      ;1              CLC
        TAX                                       BNE  BLU8      ;2              LDA  REDPNT
        PLA                              ;                                       ADC  #120
        RTI                              RED8    LDA  #$30                      STA  REDPNT
;*******************************                  STA  WSYNC                    LDA  REDPNT+1
; Display List Interrupt Routine                 STA  COLPF2                   ADC  #0
; Graphics 15, colrview                           DEX                          STA  REDPNT+1
;*******************************                  BEQ  OUT8             ;
;                                        ;                                       CLC
DLI2:   PHA                              GRN8    LDA  #$D0                      LDA  GRNPNT
        TXA                                       STA  WSYNC                    ADC  #120
        PHA                                       STA  COLPF2                   STA  GRNPNT
;                                                 DEX                           LDA  GRNPNT+1
        LDX  #192                                 BEQ  OUT8                     ADC  #0
        SEC                              ;                                       STA  GRNPNT+1
        LDA  FLAG                         BLU8    LDA  #$80             ;
        BEQ  RED02             ;0                  STA  WSYNC                    CLC
        SBC  #1                                    STA  COLPF2                  LDA  BLUPNT
        BEQ  GRN02             ;1                  DEX                          ADC  #120
        BNE  BLU02             ;2                  BNE  RED8                    STA  BLUPNT
;                                        ;                                      LDA  BLUPNT+1
                                                                                ADC  #0
```

```
;              STA BLUPNT+1
;
               INC LINCNT
               LDA LINCNT
               CMP #64
               BNE TOPP
               RTS
;
;*******************************
; ENABLE ColrView Screen
;*******************************
; save location of current
; display list and VBI routine
;
CVON:   LDA SDLSTL
        STA SAVDLS
        LDA SDLSTL+1
        STA SAVDLS+1
;
        LDA VVBLKI
        STA SAVVBI
        LDA VVBLKI+1
        STA SAVVBI+1
;
; point to custom display list
;
        LDA #LOW[DLLIST]
        STA SDLSTL
        LDA #HIGH[DLLIST]
        STA SDLSTL+1
;
        LDA #$40  ;0 for GR.15/8
        STA GPRIOR
        STA PRIOR
;
; point to VBI routine
;
        LDY #LOW[MYVBI]
        LDX #HIGH[MYVBI]
        LDA #$06 ;immediate mode
        JSR SETVBV
;
; point to DLI routine (change
; DLI1 to 2 or 3 as required)
;
        LDA #LOW[DLI1]
        STA VDSLST
        LDA #HIGH[DLI1]
        STA VDSLST+1
;
; enable VBIs and DLIs
;
        LDA #$E0
        STA NMIEN
;
        RTS          ;that's all!
;
;*******************************
; DISABLE ColrView Screen
;*******************************
CVOFF:  LDA #$20 ;avoid changing
        STA NMIEN;during a VBI
;
```

ERRATA

A couple pesky typos snuck into the 600XL Super Video upgrade in the December '93 *AC*. Obviously the work of a bleary-eyed editor staying up *way* past his bedtime—it's amazing how numbers and letters all blur together at 3AM.

Both errors occur on page 10 of the December '93 issue. In the parts list at the top of the right-hand column, change location "L13" to read "L12". This is where the CV Disable switch is installed. The inductor at location L13 is already installed on the board and should be left in place.

Also, in Figure 8 at the bottom left-hand column, R134 was inadvertently omitted from the drawing. It should be inserted between R132 and R133 in the board diagram. Our thanks to Decker McAllister of Seal Beach, CA for bringing these items to our Attention. *AC* regrets any inconvenience.

Unicorn Publications
3487 Braeburn Cir.
Ann Arbor, MI 48108